

# **Projet Mobile**

Docly : Application Mobile pour la Gestion des Rendez-vous et  
Consultations Médicales

## **Auteurs :**

Adem Ali Sayadi

Anis Chouchene

Maâroufi Med Hedi

**Section :** 2GInfo

**Date de Soumission :** 07 Mai 2025

**Université de Tunis, École Nationale Supérieure  
d'Ingénieurs de Tunis (ENSIT)**

---

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation de l'Application, Fonctionnalités et Cas d'Utilisation</b>	<b>2</b>
2.1	Fonctionnalités Principales	2
2.2	Cas d'Utilisation	2
<b>3</b>	<b>Modules et Fonctionnalités</b>	<b>3</b>
3.1	Modules Frontend (IONICPROJECT)	3
3.2	Modules Backend (BACKEND)	3
<b>4</b>	<b>Interactions des Modules</b>	<b>4</b>
4.1	Authentification	4
4.2	Gestion du Profil	4
4.3	Prise de Rendez-vous	4
4.4	Gestion des Consultations	5
4.5	Gestion Administrative	5
<b>5</b>	<b>Captures d'Écran</b>	<b>5</b>
<b>6</b>	<b>Manuel d'Utilisation</b>	<b>5</b>
6.1	Pour les Patients	5
6.2	Pour les Médecins	5
6.3	Pour les Admins	5
<b>7</b>	<b>Considérations Techniques</b>	<b>6</b>
<b>8</b>	<b>Exemples de Code</b>	<b>7</b>
8.1	Frontend (HTML/Angular)	7
8.2	Backend (Python/Flask)	7
<b>9</b>	<b>Conclusion</b>	<b>7</b>
<b>10</b>	<b>Annexes</b>	<b>7</b>
10.1	Ressources	7

# 1 Introduction

Ce document présente "Docly", une application mobile hybride développée dans le cadre d'un mini-projet à l'ENSIT pour la section 2GInfo par Adem Ali Sayadi, Anis Chouchene et Maâroufi Mohamed Hedi. L'application vise à optimiser la gestion des cabinets médicaux en facilitant les interactions entre patients, médecins et administrateurs. Construite avec Ionic 6+ et Angular pour le frontend, Python Flask avec MongoDB la base de données, et Render.com pour le déploiement du backend, "Docly" cible les plateformes mobile. Ce rapport détaille les fonctionnalités, les cas d'utilisation, la structure des modules, leurs interactions, les considérations techniques, et fournit un manuel d'utilisation complet.

Le projet répond au besoin d'une gestion efficace des soins de santé en permettant aux patients de planifier des rendez-vous, de téléverser des documents médicaux et de consulter leur historique, tout en offrant aux médecins la gestion des consultations, l'analyse de leur pratique et l'interaction avec les patients. Un rôle d'administrateur assure un contrôle d'accès sécurisé en gérant les comptes des médecins. Ce rapport offre un aperçu technique détaillé et des instructions pour les utilisateurs.

## 2 Présentation de l'Application, Fonctionnalités et Cas d'Utilisation

Cette section décrit l'objectif du projet, ses fonctionnalités principales et les cas d'utilisation typiques.

### 2.1 Fonctionnalités Principales

Liste des fonctionnalités clés :

- **Création de compte et authentification** : Inscription et connexion sécurisée via email et mot de passe.
- **Gestion des Rendez-vous** : Recherche, réservation et gestion par patients et médecins.
- **Gestion des Documents (PDF)** : Téléversement et annotation de documents médicaux.
- **Suivi des Consultations** : Enregistrement des détails par les médecins.
- **Recherche et Planification** : Filtrage par spécialité et réservation.
- **Contrôles Administratifs** : Approbation/rejet des comptes médecins.
- **Analytique** : Statistiques pour les médecins.

### 2.2 Cas d'Utilisation

Scénarios typiques :

- **Patients** :
  - *Inscription* : Un patient s'inscrit via un formulaire.
  - *Réservation* : Recherche un neurologue et réserve pour le 9 mai 2025.
  - *Téléversement* : Soumet un rapport médical.

- *Historique* : Consulte une visite passée (14 juillet 2024).
- *Mise à jour* : Modifie ses informations personnelles.
- **Médecins** :
  - *Gestion* : Confirme ou annule des rendez-vous.
  - *Consultation* : Enregistre un diagnostic et une ordonnance.
  - *Analytique* : Vérifie les statistiques mensuelles.
- **Administrateurs** :
  - *Gestion* : Approuve ou rejette un compte médecin.

## 3 Modules et Fonctionnalités

Décomposition en modules.

### 3.1 Modules Frontend (IONICPROJECT)

Structure avec Ionic 6+ :

- **Module App** : Initialisation, dépendances, routes.
- **Module Pages** : home, login, signup-options, signup-form, patient-profile, doctor-profile, admin-dashboard, etc.
- **Module Models** : Interfaces TypeScript (User, Appointment, Doctor).
- **Module Services** : Requêtes HTTP (ex. /api/auth/login).
- **Modules Assets et Theme** : Ressources et styles.

### 3.2 Modules Backend (BACKEND)

Structure avec Flask et MongoDB :

- **Module Models (models/)** :
  - *Fonctionnalités* : User, Patient, Doctor, Admin, Appointment, Consultation.
  - *Fichiers* : user.py, patient.py, doctor.py, admin.py, appointment.py, consultation.py.
- **Module Routes (routes/)** :
  - *Fonctionnalités* :
    - Auth (auth.py, préfixe /api/auth) :
      - POST /register : Inscription utilisateur.
      - POST /login : Authentification et génération de token JWT.
    - Patient (patient.py, préfixe /api/patient) :
      - POST /appointment : Réservation de rendez-vous.
      - POST /document : Téléversement de document.
    - Doctor (doctor.py, préfixe /api/doctor) :
      - GET /appointments : Liste des rendez-vous.
      - GET /patients : Liste des patients.
    - Admin (admin.py, préfixe /api/admin) :

- POST /status : Mise à jour du statut médecin.
- Service (service.py, préfixe /api/service) :
  - GET /list : Liste des services médicaux.
- Notification (notification.py, préfixe /api/notification) :
  - POST /send : Envoi de notifications.
- Search (search.py, préfixe /api/search) :
  - GET /doctors : Recherche de médecins.
- Health Check (run.py, préfixe /) :
  - GET / : Vérification de l'état de l'API.
- *Fichiers* : auth.py, patient.py, doctor.py, admin.py, service.py, notification.py, search.py.
- **Module Utils (utils/)** :
- *Fonctionnalités* : Configuration (Config), initialisation MongoDB (init\_mongo), seedingadmin(seed\_static\_admin). *Fichiers* : config.py, database.py.
- **Point d'Entrée (run.py)** :
  - *Fonctionnalités* : Initialise Flask, configure CORS, JWT, et MongoDB, enregistre les blueprints, gère les erreurs 404/500.
  - *Fichiers* : run.py.

## 4 Interactions des Modules

Détail des interactions.

### 4.1 Authentification

- **Frontend** : Pages (login/signup), Services (POST /api/auth/register).
- **Backend** : Routes (auth.py), Models (user.py).
- **Flux** : Inscription → Sauvegarde → ID. Connexion → Token JWT.

### 4.2 Gestion du Profil

- **Frontend** : Pages (profile), Services (GET /api/user/profile).
- **Backend** : Routes (user.py), Models (user.py).
- **Flux** : Demande → Données. Mise à jour → Sauvegarde.

### 4.3 Prise de Rendez-vous

- **Frontend** : Pages (booking), Services (POST /api/patient/appointment).
- **Backend** : Routes (patient.py), Models (appointment.py).
- **Flux** : Sélection → Réservation → Sauvegarde.

## 4.4 Gestion des Consultations

- **Frontend** : Pages (consultation), Services (POST /api/doctor/consultation).
- **Backend** : Routes (doctor.py), Models (consultation.py).
- **Flux** : Création → Sauvegarde → ID.

## 4.5 Gestion Administrative

- **Frontend** : Pages (admin-dashboard), Services (POST /api/admin/status).
- **Backend** : Routes (admin.py), Models (doctor.py).
- **Flux** : Mise à jour → Sauvegarde.

# 5 Captures d'Écran

Preuves visuelles :

# 6 Manuel d'Utilisation

Instructions détaillées.

## 6.1 Pour les Patients

1. **Inscription** : Choisissez "S'inscrire" et remplissez le formulaire.
2. **Connexion** : Connectez-vous avec email et mot de passe.
3. **Réservation** : Allez à "Accueil", sélectionnez un service.
4. **Téléversement** : Dans "Profil" > "Dossier", téléversez.
5. **Historique** : Consultez dans "Profil".
6. **Mise à Jour** : Modifiez dans "Profil".

## 6.2 Pour les Médecins

1. **Inscription** : S'inscrire, attendre approbation.
2. **Connexion** : Connectez-vous.
3. **Patients** : Consultez dans "Patients".
4. **Consultations** : Ajoutez des notes.
5. **Analytique** : Vérifiez les stats.
6. **Téléchargement** : Téléchargez via "Télécharger".

## 6.3 Pour les Admins

1. **Connexion** : Utilisez identifiants admin.
2. **Gestion** : Consultez "Admin Dashboard".
3. **Mise à Jour** : Changez statut, confirmez.

## 7 Considérations Techniques

Aspects techniques :

- **API** : Endpoints RESTful avec JSON, codes HTTP (200, 404, 500). Utilisation de blueprints Flask pour modularité (ex. /api/auth, /api/patient).
- **Sécurité** : Authentification via JWT (Flask-JWT-Extended), CORS configuré pour restreindre les origines, support des cookies.
- **Base de Données** : MongoDB initialisé avec PyMongo, seeding d'un admin statique au démarrage.
- **Évolutivité** : Modules paresseux frontend, Flask sans état, blueprints pour modularité backend.

## 8 Exemples de Code

### 8.1 Frontend (HTML/Angular)

```
<ion-tab tab="appointments">
  <div id="appointments-page" class="content">
    <app-appointments></app-appointments>
  </div>
</ion-tab>
```

### 8.2 Backend (Python/Flask)

```
from flask import Blueprint, jsonify

auth_bp = Blueprint('auth', __name__)

@auth_bp.route('/register', methods=['POST'])
def register():
    # Logique d'inscription
    return jsonify({"message": "Utilisateur inscrit"}), 201
```

## 9 Conclusion

Cette Application offre une solution robuste pour la gestion des soins, avec une architecture modulaire. Les améliorations futures incluent OAuth et analytique avancée.

## 10 Annexes

### 10.1 Ressources

Liens vers GitHub et guides : [github.com](https://github.com)



# Docly

## Services [Show More >](#)



Neuro



General

## Doctors [Show More >](#)

MH

**Mohamed Hedi Maaroufi**

Neuro

DG

**David Green**

General

FM

**Frank Miller**

General



Home



Appointments



Search



Notification



Profile

FIGURE 1 – Écran d'accueil

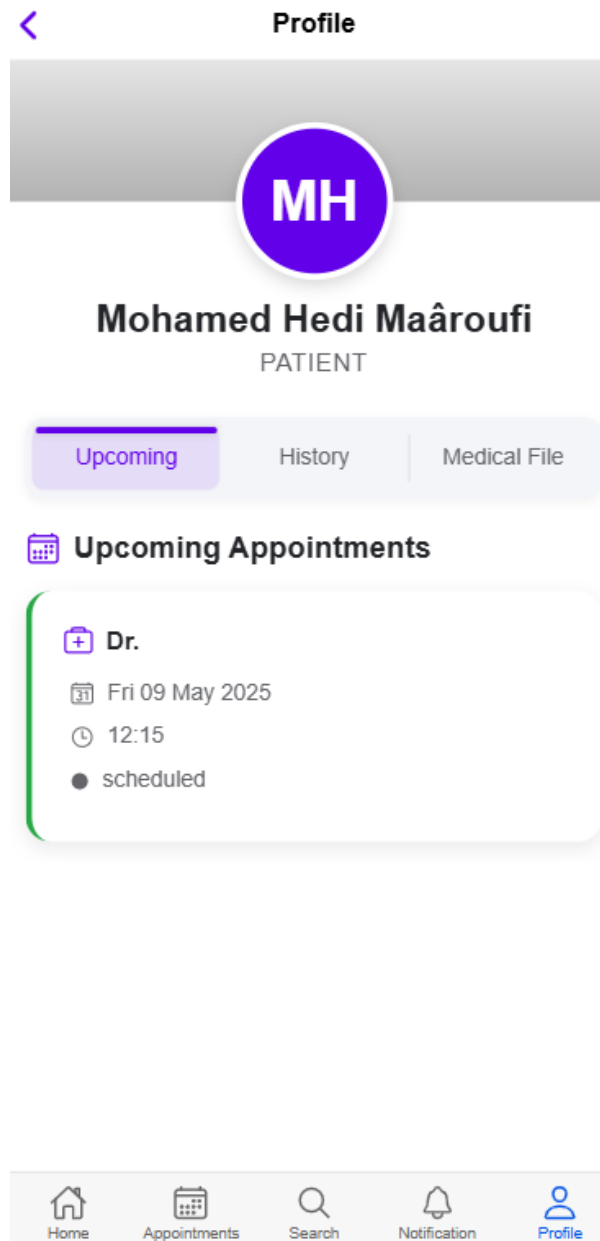


FIGURE 2 – Profil patient - Rendez-vous à venir

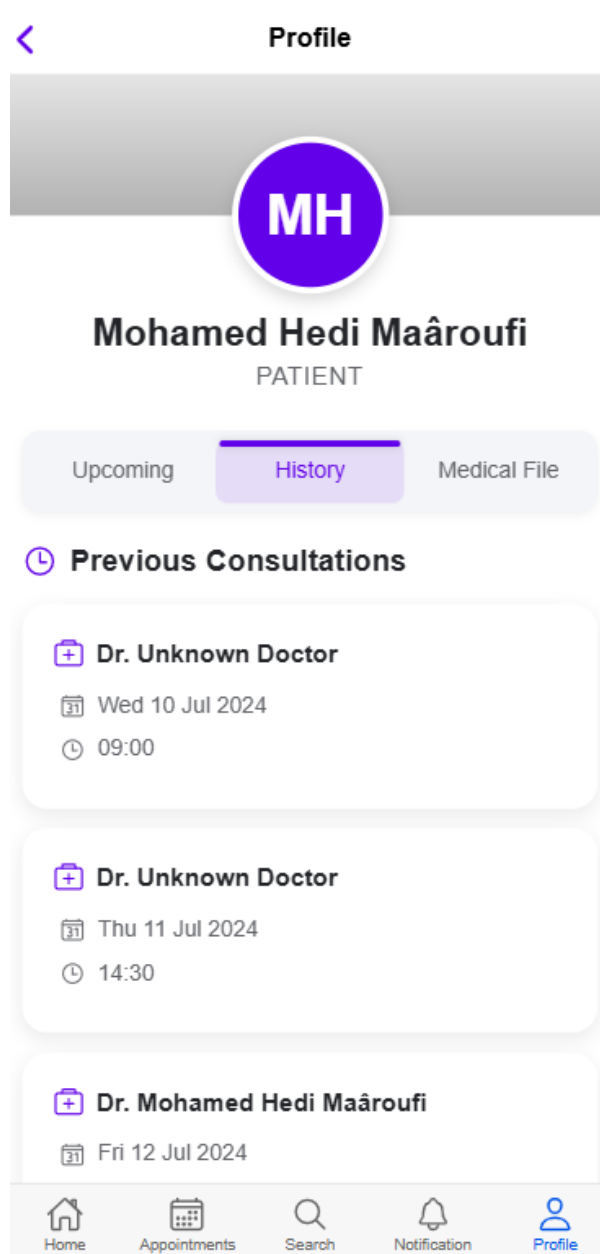


FIGURE 3 – Profil patient - Historique

Profile


Mohamed Hedi Maâroufi


PATIENT

Upcoming

History

Medical File

 Medical File



Your Medical Records

Access your complete medical history

Download

Select Document

Choose file


No file chosen


Description


Enter document description


UPLOAD

RESET

 Home

 Appointments

 Search

 Notification


 Profile

FIGURE 4 – Profil patient - Téléversement

11

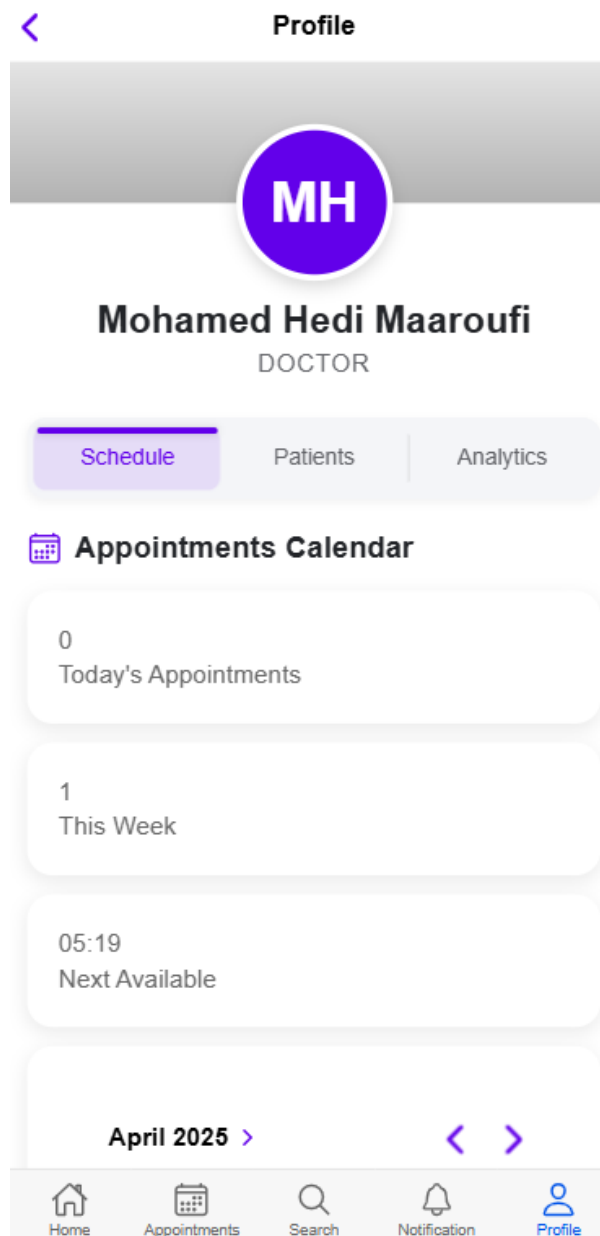


FIGURE 5 – Profil médecin - Calendrier

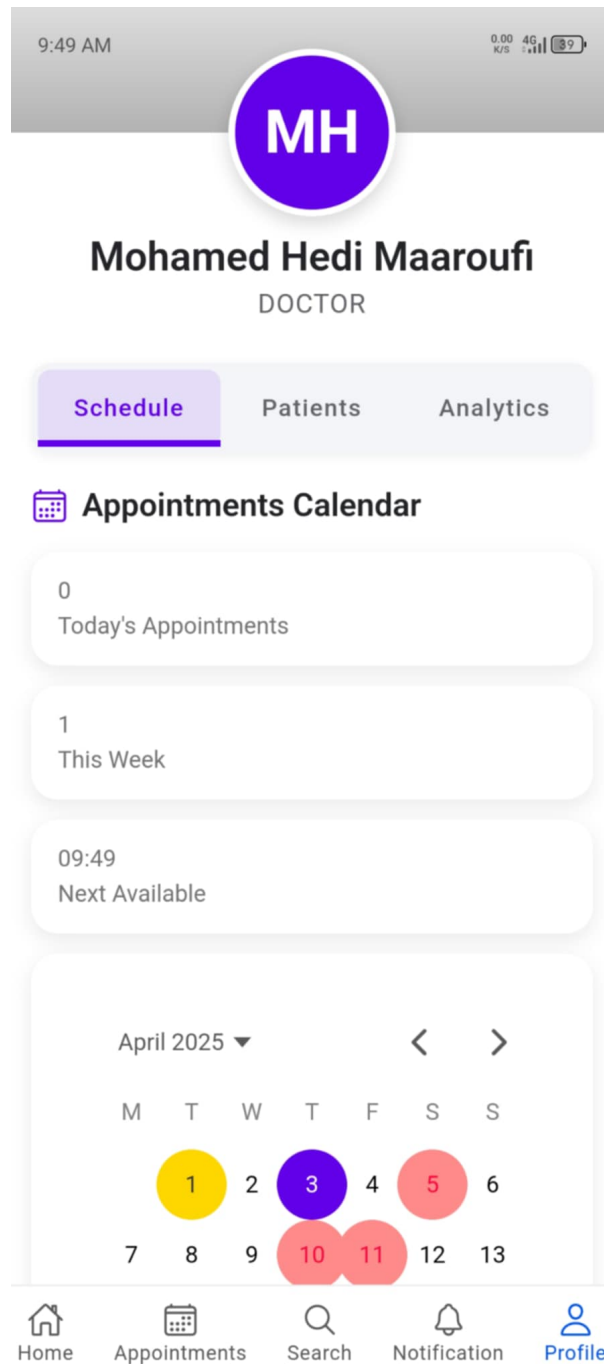


FIGURE 6 – Profil médecin - Calendrier

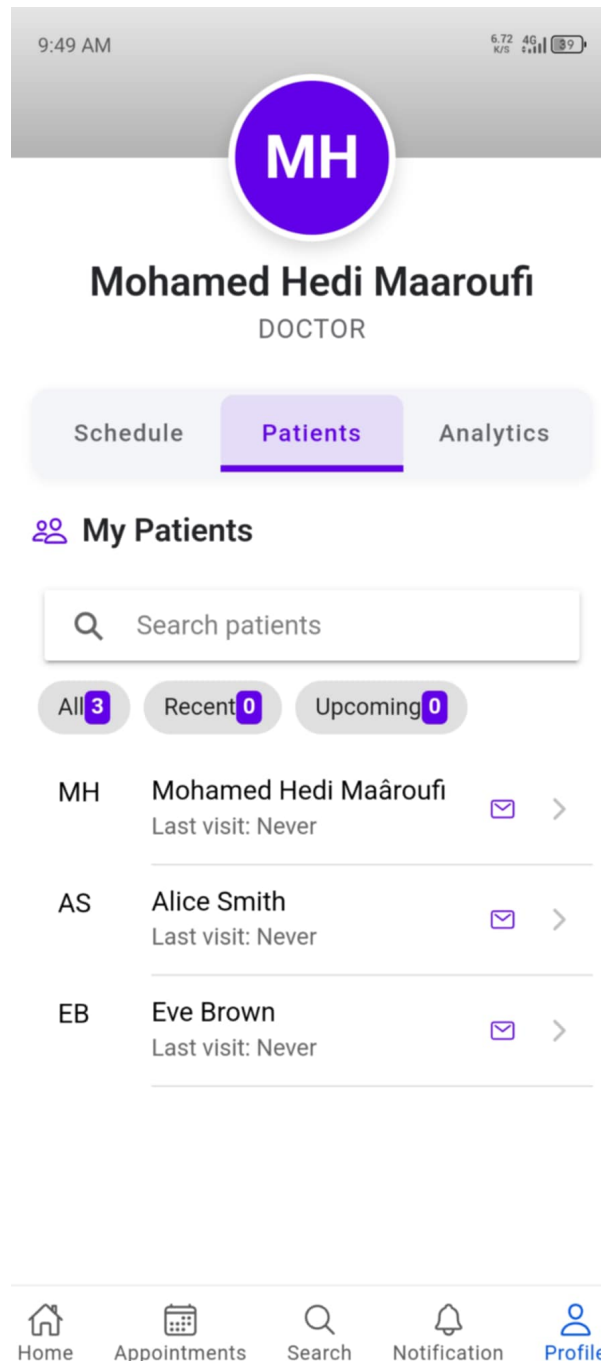


FIGURE 7 – Gestion des patients

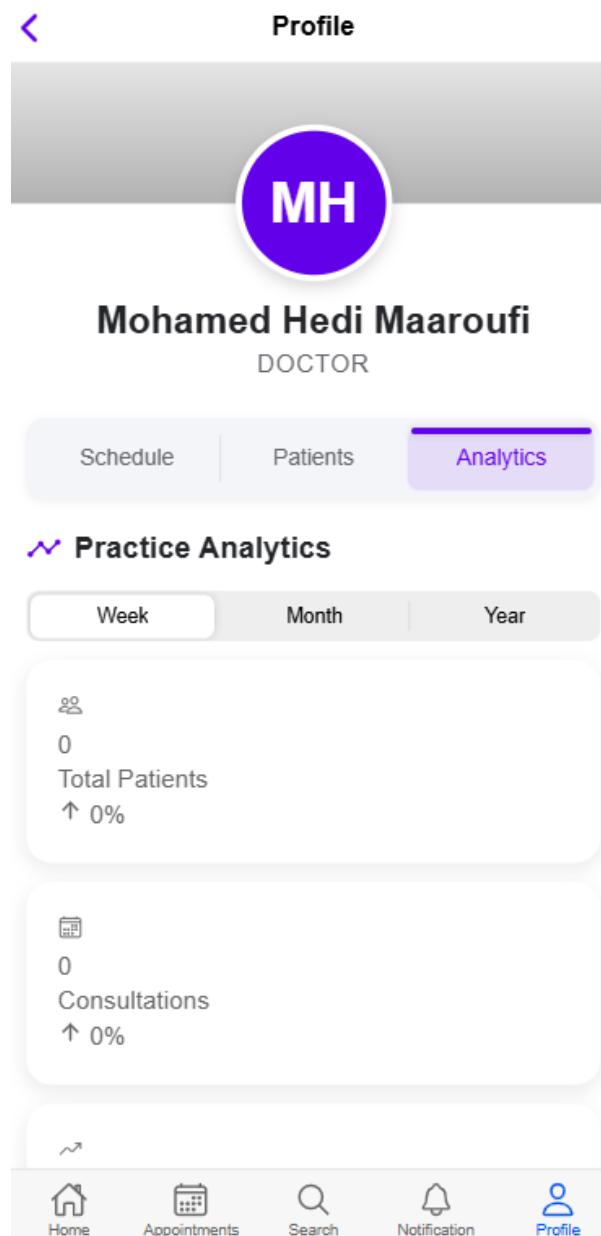


FIGURE 8 – Tableau analytique



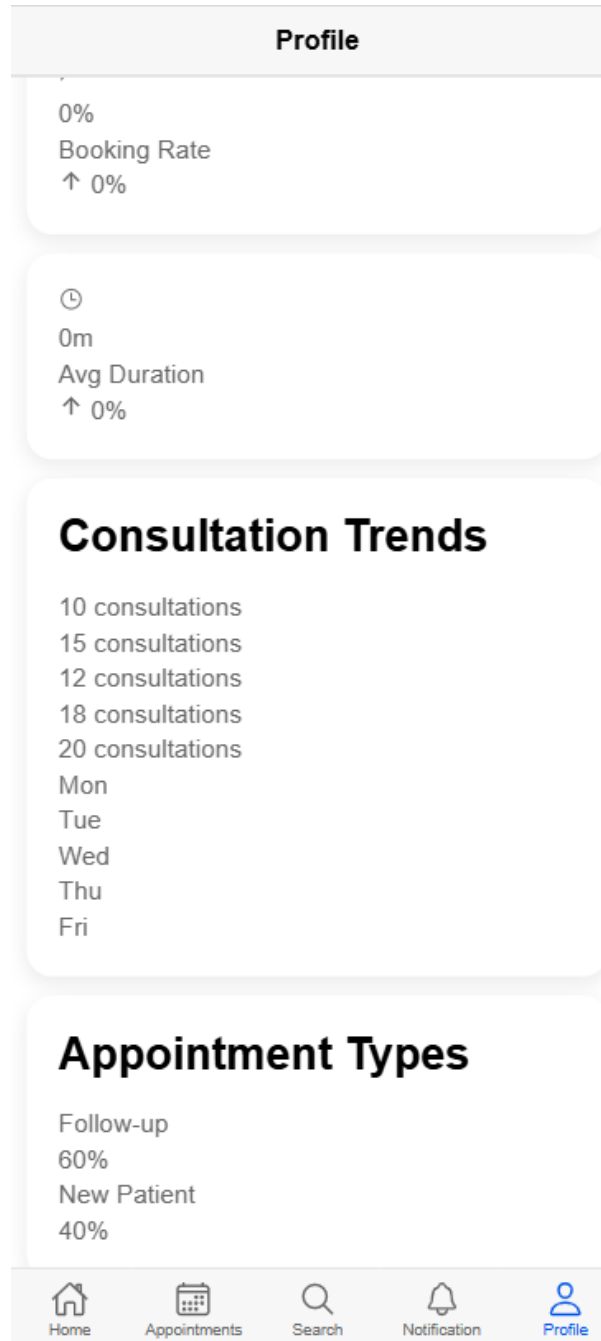





FIGURE 9 – Tableau analytique


**Doctor Profile** ×



**Dr. Mohamed Hedi Maaroufi**  
Neuro

 mohamedhedi.maaroufi@ensit.u-tunis.tn

 Grombalia

 54544545444


**Book Appointment** 

FIGURE 10 – Détails profil médecin

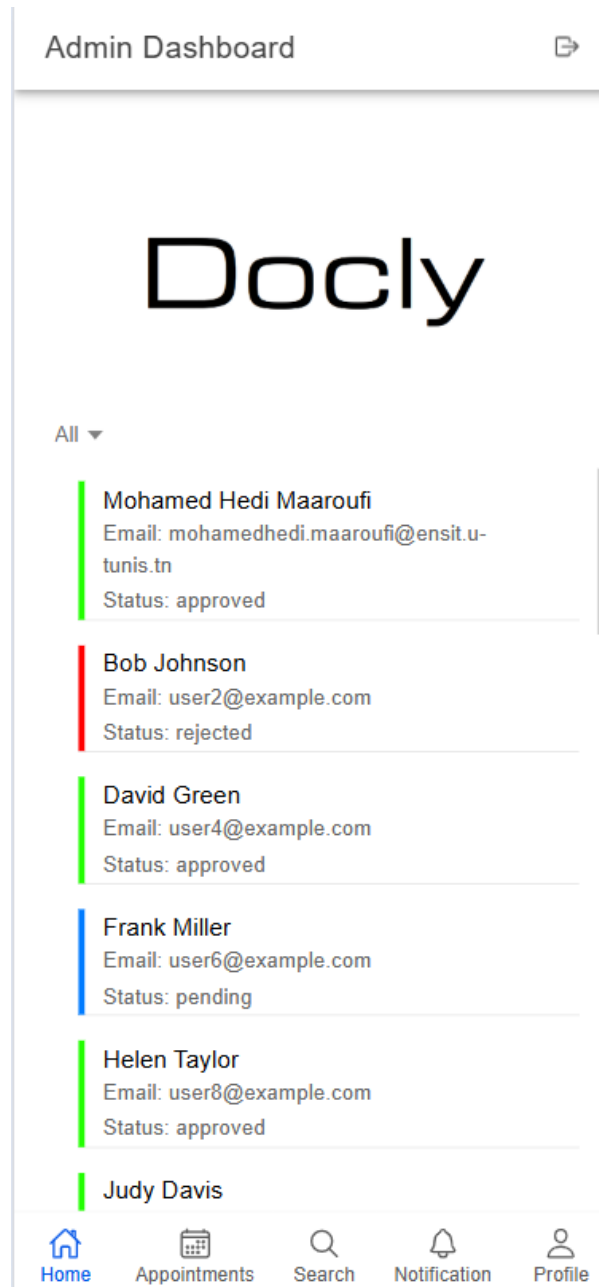


FIGURE 11 – Tableau de bord administrateur

[← Back](#)

# Docly

Email

Password

**LOG IN**

Don't have an account? [SIGN UP](#)

FIGURE 12 – Écran de connexion

[< Back](#)

# Docly

**SIGNUP AS PATIENT**

**SIGNUP AS DOCTOR**

Already have an account? **LOG IN**

FIGURE 13 – Sélection de rôle

[< Back](#)

# Docly

Name

Email

Phone

Address

Date of Birth

Select date

Password

**SIGN UP**

Already have an account? [\*\*LOG IN\*\*](#)

FIGURE 14 – Formulaire d'inscription client

[< Back](#)

# Docly

Name

Speciality

Email

Phone

Address

Date of Birth

Select date

Password

**SIGN UP**

FIGURE 15 – Formulaire d'inscription docteur