**Nom & Prénom : NAÏMATOU SIMPORE**

**Numéro Etudiant : 21921358**

**MASTER 2 DATASCALE**

**Compte-Rendu du module de gestion de données et services dans le cloud**

**7°)** Création d’une image Docker déployant une application Web donné sur un serveur apache

Dans cette partie, nous allons créer une image déployant une application web sur un serveur d’apache, nous allons la publier puis nous l’instancierons en créant un ou plusieurs conteneurs.

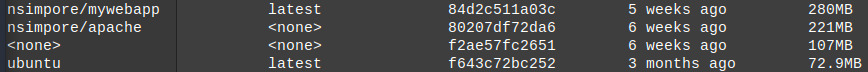
* **Le Dockerfile**

On crée un dossier dans lequel on crée un fichier Dockerfile qui va contenir les différents paramètres permettant de créer l’image docker.

****

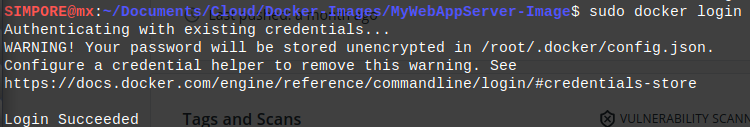
* **Vérification de la création**

Après exécution, on peut vérifier que notre image a bien été créée ce qui est le cas (nsimpore/mywebapp):

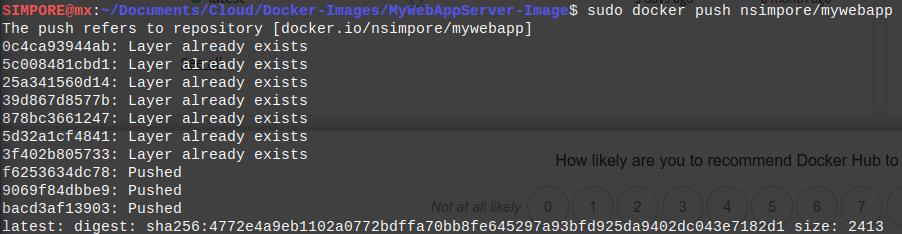


* **Publication de l’image docker sur Dockerhub**

Pour ce faire, on se logue comme représenter sur l’image ci-dessous :

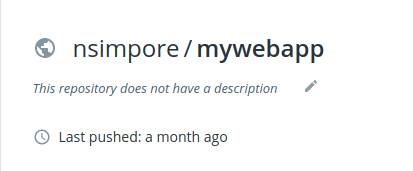


Puis à l’aide de la commande docker push on publie notre image en prenant le soin de préciser son nom :



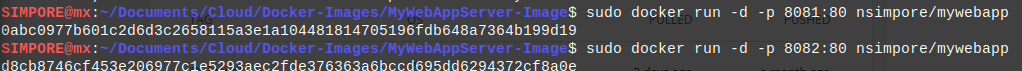
* **Vérification du dépôt sur Dockerhub**

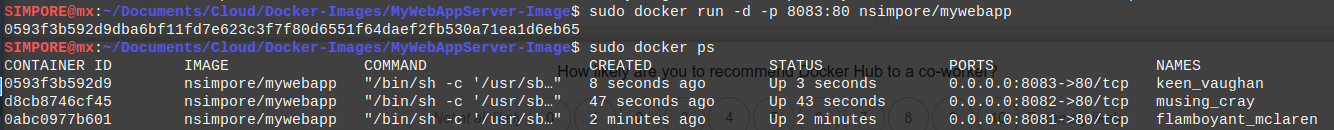
Après vérification, nous constatons bien que notre image a bien été pusher



* **Instanciation de notre image**

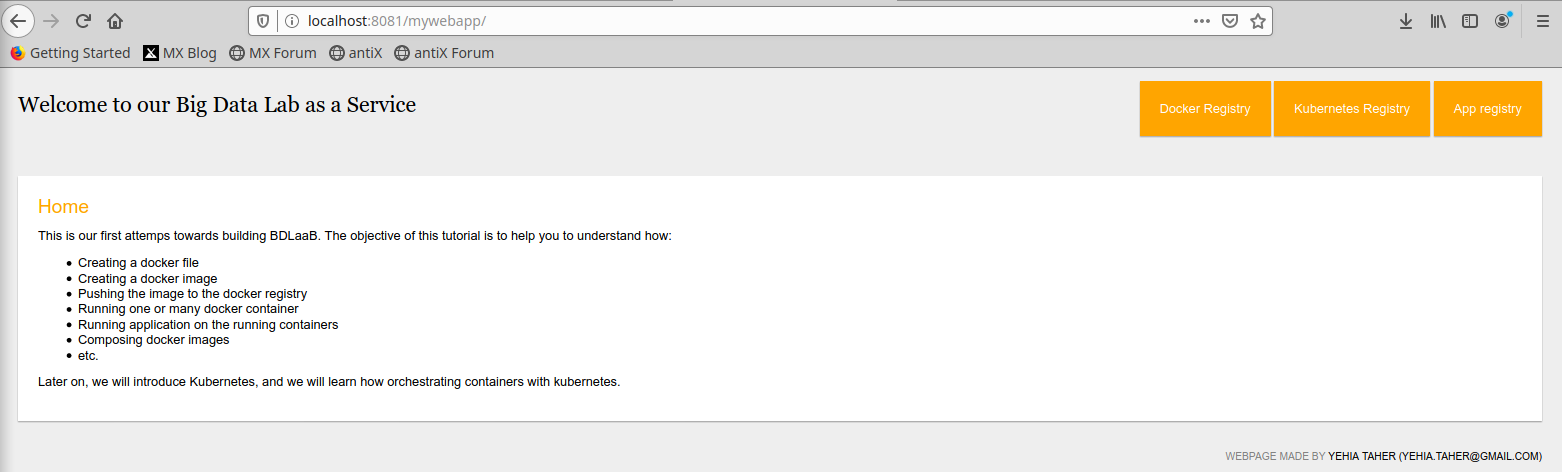
Instancions notre image en créant plusieurs conteneurs déployant notre application. Pour ce faire, nous avons créé 3 conteneurs en renseignant à chaque fois des ports différents : 8081, 8082, 8083 comme le montre les images ci-dessous.





* **Test de l’application**

Pour tester notre application, notre navigateur web à l’adresse <http://localhost:8081/mywebapp>

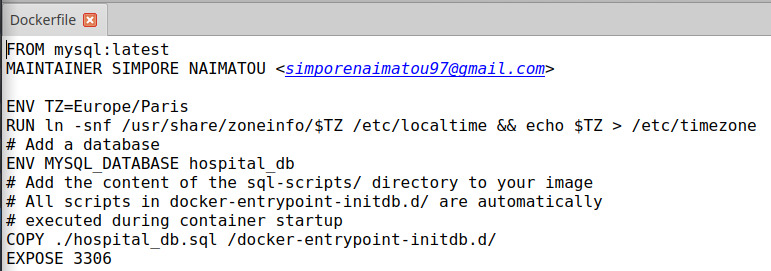


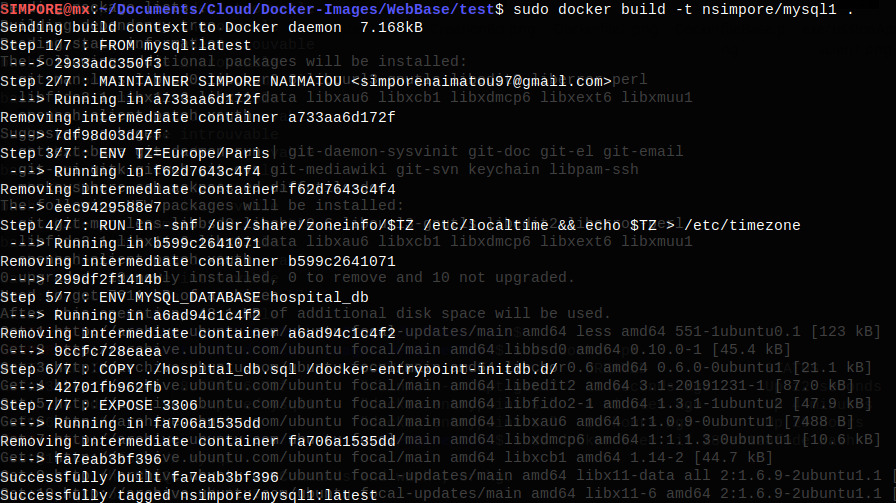
8°) Création d’une image docker déployant une application web donnée sur un serveur apache

Dans cette partie, nous allons créer 2 conteneurs : dans un premier temps nous créons une image mysql et apache, puis daux conteneurs un pour mysql et un autre pour apache. Pour la création du conteneur d’apache nous établissons le lien vers la base de donnée mysql à travers le conteneur que nous avons créé.

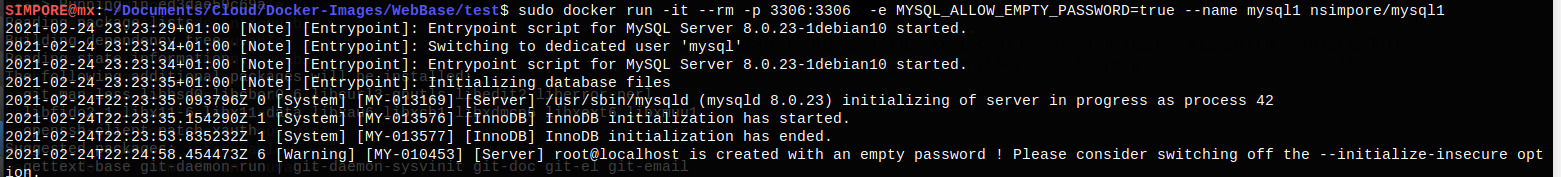
* **Image Mysql**

On commence par créer un nouveau dossier, puis en me plaçant sur ce dossier, on crée un fichier Dockerfile dans lequel on spécifie les paramètres à prendre en compte en ce qui concerne les mises à jour mais aussi les installations comme nous montrent l’image ci-dessous :

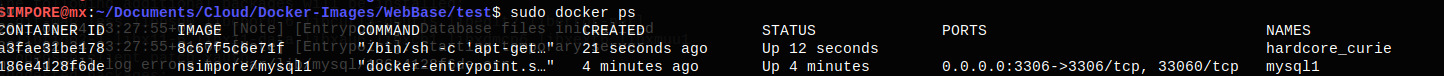
****

Après on exécute le fichier : 

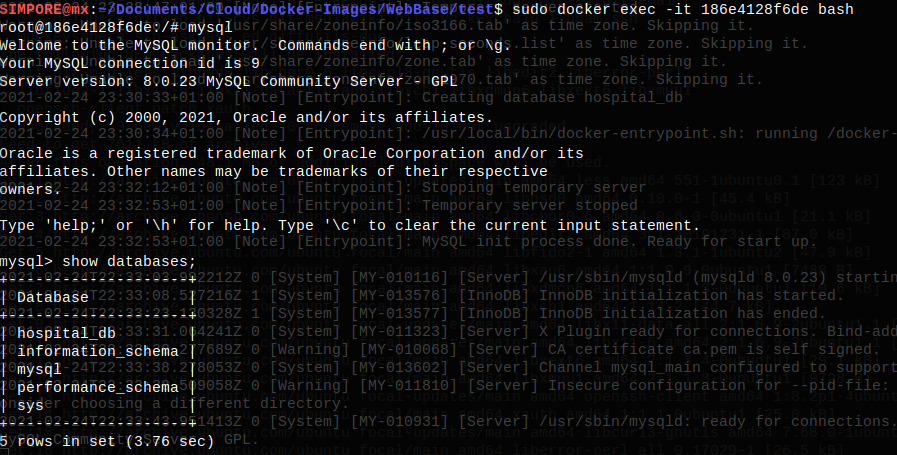
Après création de l’image, on crée un conteneur pour l’image comme nous montre l’image ci-dessous :

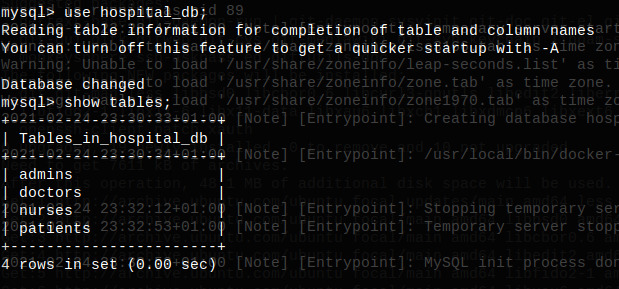


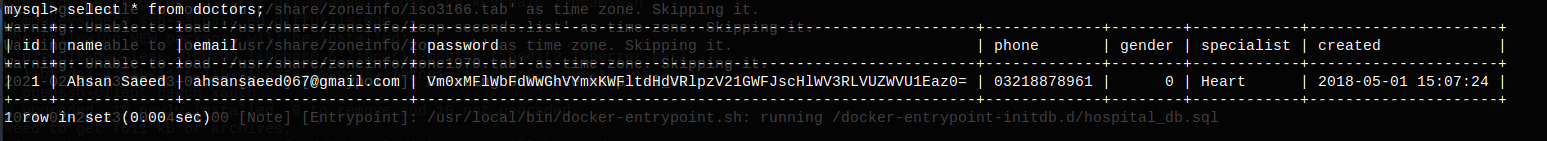
On peut vérifier que notre conteneur a bien été créé :



On effectue quelques requêtes pour s’assurer du bon déroulement, mais avant on se place sur le conteneur :

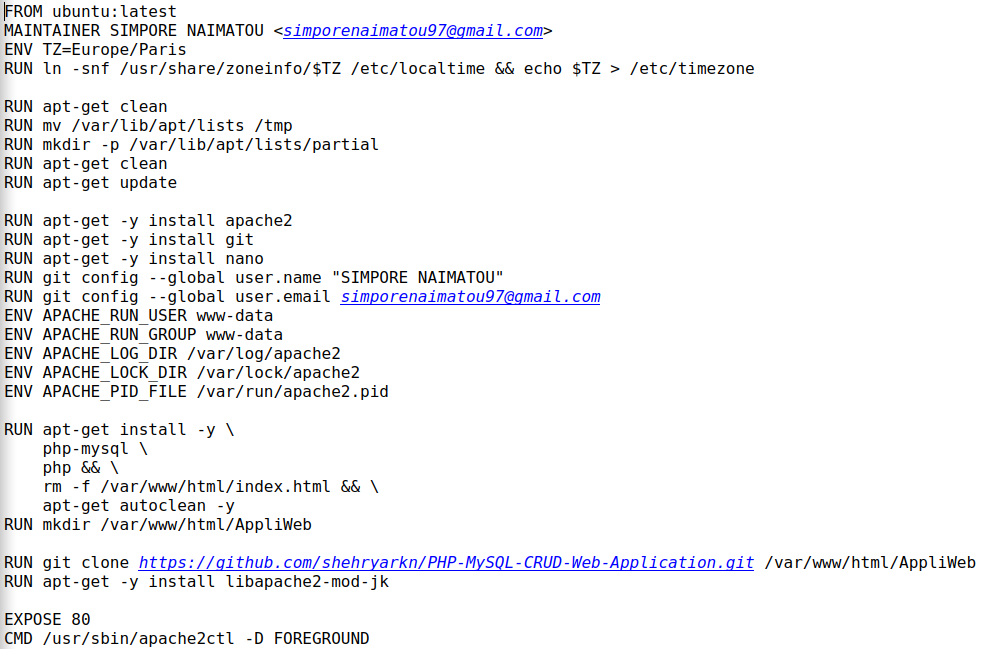






* **Image apache**

On commence par créer un nouveau dossier, puis en se plaçant sur ce dossier, on crée un fichier Dockerfile dans lequel on spécifie les paramètres à prendre en compte en ce qui concerne les mises à jour mais aussi les installations. Pour ce faire, nous avons utilisé une ancienne image ubuntu comme nous montre l’image ci-dessous.

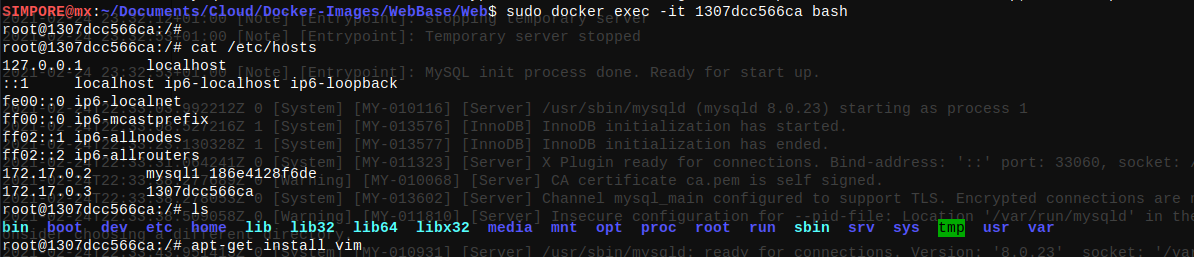
****

Après l’exécution du fichier, on lie l’image du conteneur apache à celui de Mysql à travers la commande : C:\Users\utilisateur\Downloads\1.png

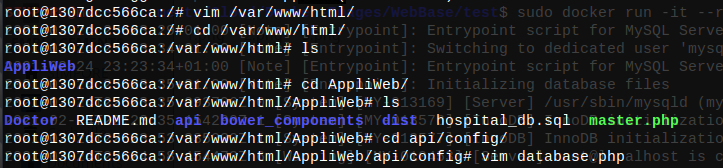
On vérifie que nos conteneurs ont bien été créé :

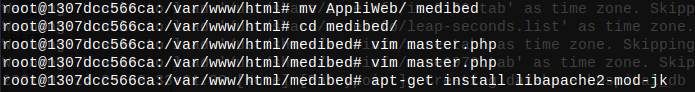


Puis en se plaçant sur ce conteneur, en prenant le soin de vérifier notre connexion :

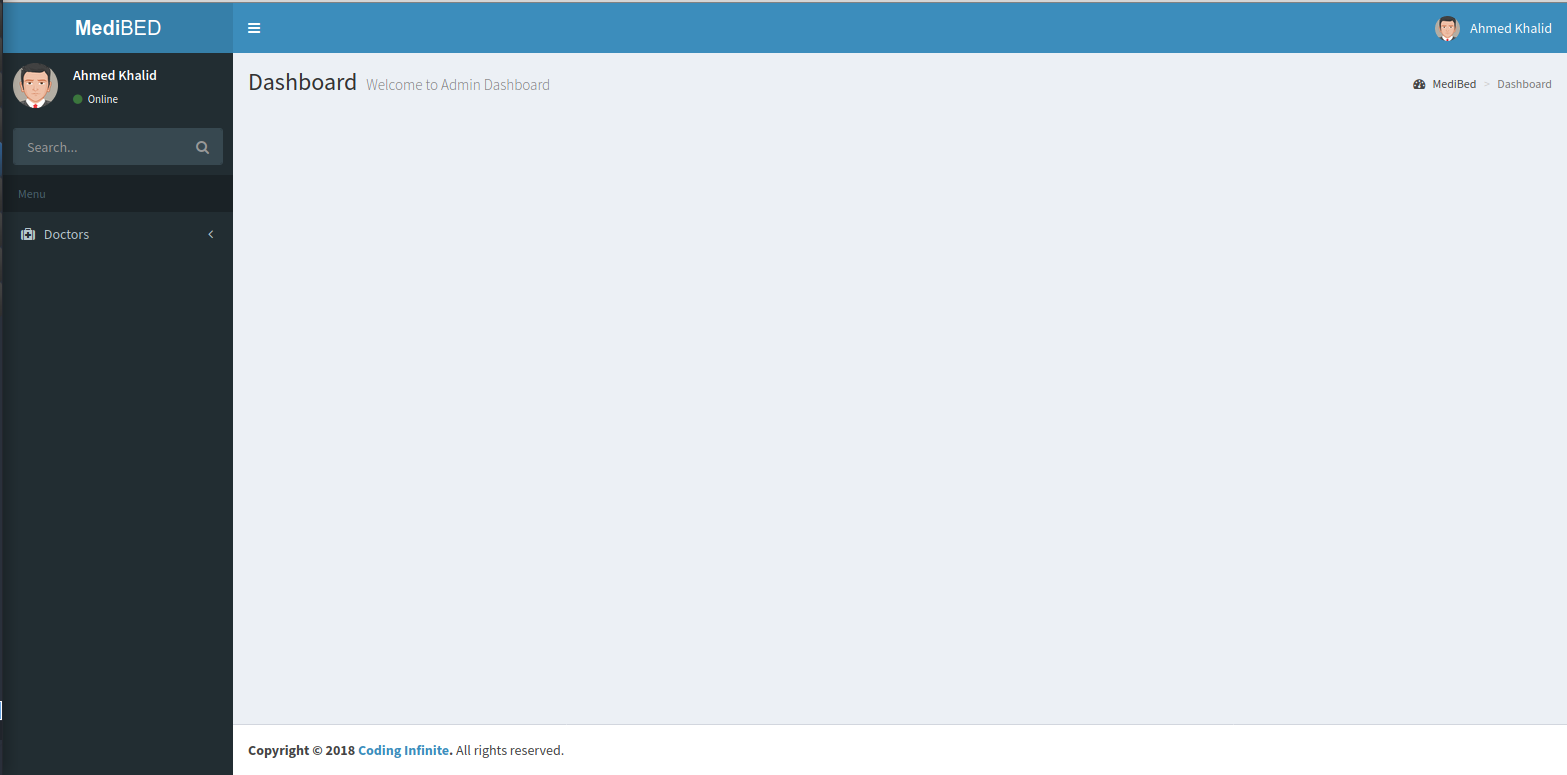


On édite le fichier database.php en prenant le soin de modifier le nom de la base de données et aussi le fichier master.php en spécifiant les bons chemins afin de pouvoir y avoir accès sur le navigateur:

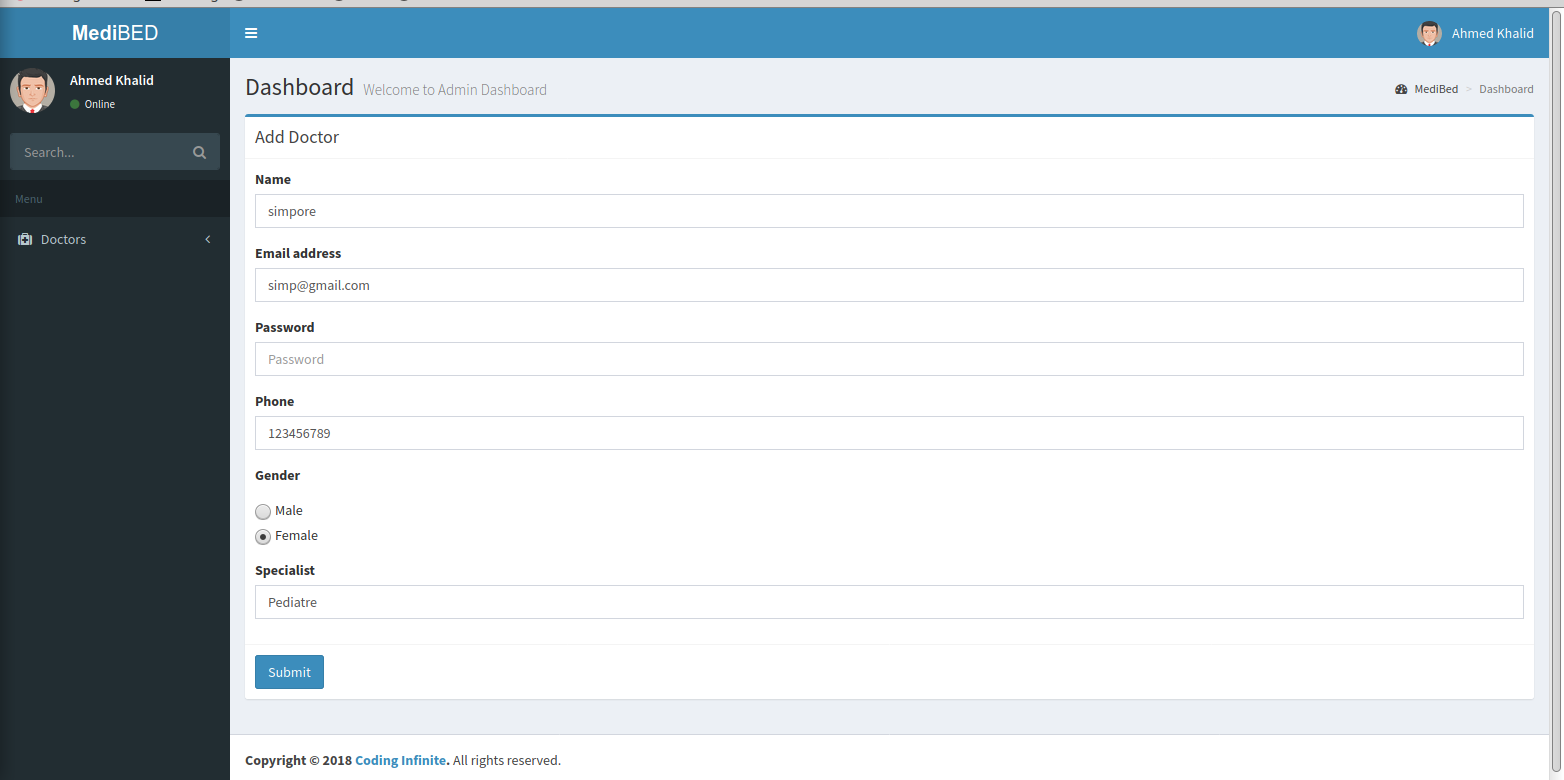




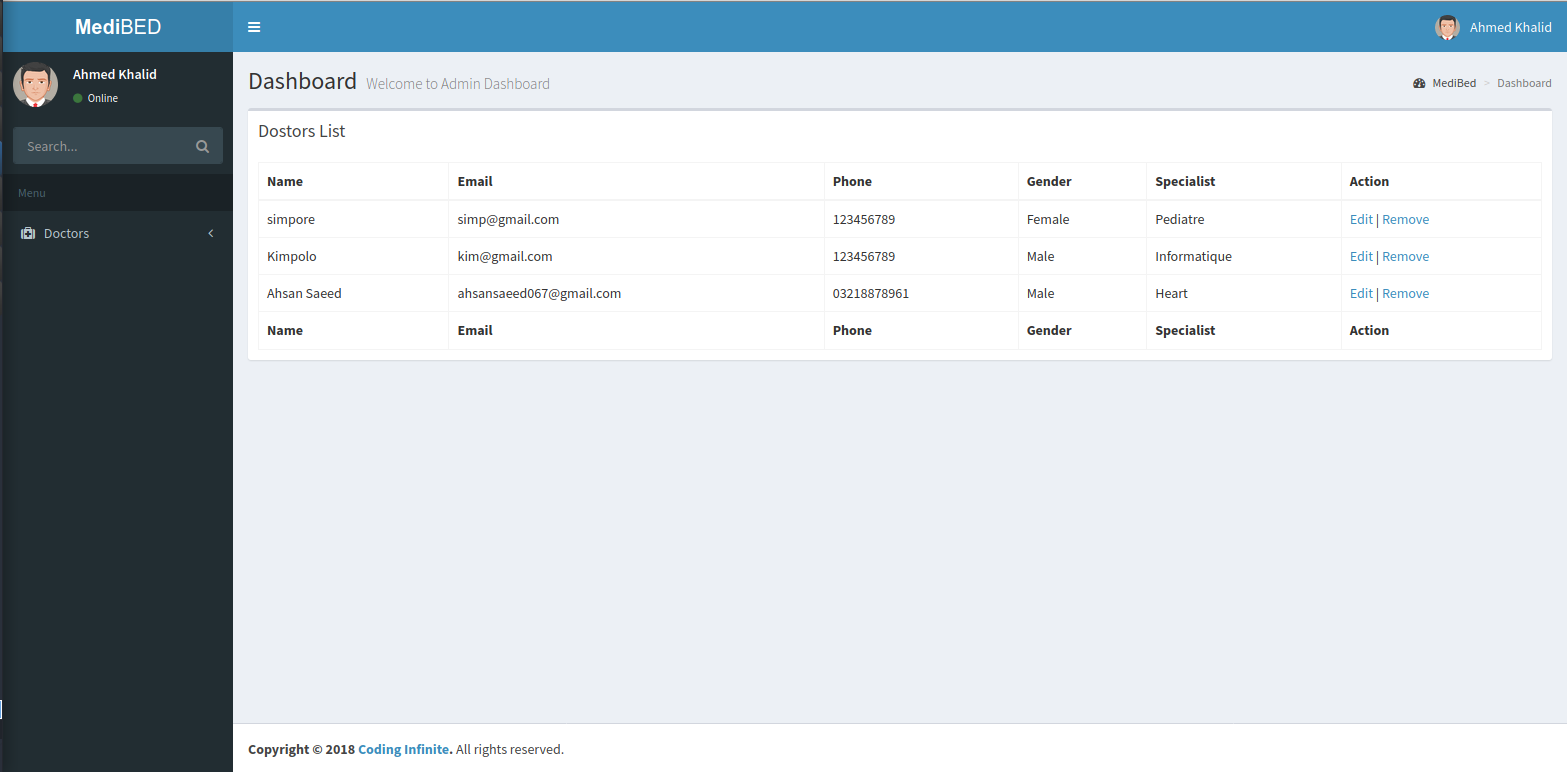
Pour s’assurer que tout s’est bien passé, on se place sur le navigateur au port indiqué pour avoir accès à l’application :



On peut faire des tests d’insertion de nouveaux docteurs comme nous montre l’image ci-dessous :



La liste des docteurs :



**9°) Opérateur de composition de service**

Dans cette partie, nous allons créer un opérateur de composition de service SaaS de type application et de service PaaS de type serveur apache. Pour ce faire, nous avons créé un programme permettant d’automatiser tout le processus.

* **Description**