الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة 1
Université SAAD DAHLAB de BLIDA1

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Télécommunication
Spécialité Réseaux & Télécommunications
Présenté par

ZIANE Mohamed Riadh
&
AKILI Ali

Réalisation d'une plateforme de gestion d'un réseau de capteurs

Proposé par : Mr KABIR Yacine

Année Universitaire 2021-2022

Remerciements

En premier lieu, nous remercions Dieu le tout puissant de nous avoir donné le privilège et la chance d'étudier et de suivre le chemin de la science et de la connaissance.

En second lieu, nos remerciements les plus sincères vont droit à nos parents qui n'ont pas cessé de nous encourager et nous soutenir tout au long du long parcours que furent nos études.

Ce travail était effectué sous l'encadrement de monsieur KABIR Yacine docteur à la faculté d'électronique de Saad Dahleb Blida 1. Toute notre gratitude à notre encadreur de mémoire, Dr KABIR, avec qui nous avons eu le plaisir de travailler.

Nous le remercions vivement d'avoir assuré la direction et le bon déroulement de ce travail ainsi que pour ses conseils précieux.

Sans oublier toutes les personnes qui ont participé de près ou de loin pour achever ce travail

Enfin, nous remercions nos chers amis qui nous ont soutenus dans tous nos parcours du cycle Des études

Dédicaces

Ce travail est spécialement dédié à :

Mes chers parents, qui ont consacrés leurs existences à bâtir la mienne, pour leur soutient, patience et soucis de tendresse et d'affection, pour tous ce qu'ils ont fait pour que je puisse arriver à ce stade.

A ma sœur et mon frère. Je ne cesserai de témoigner mon amour envers vous.
-A mon cher ami et mon binôme Riadh.
-A mes chers amis qui m'ont soutenu et encouragés.
-A tous ce qui m'ont aidé de près ou de loin.

AKILI Ali

Je dédie ce modeste travail
- A ma très chère mère, source de vie, d'amour et d'affection,
- A toute ma famille source d'espoir et de motivation,
- A tous mes amis et mon cher ami avant d'être binôme,
- A tous ceux qui m'ont aidé de près ou de loin

ZIANE Mohamed Riadh

Abstract

The Internet of Things technology connects the physical world to the digital world. Its architecture includes a hardware part (microcontrollers) and a software part (platform, application...). The aim of this project is to create a platform that allows the collection of data from a network of sensors, but also to store this data for future analysis or studies. We hope, with the results obtained, to contribute to the improvement and generalization of this technology in our country.

Keywords : Internet of things, microcontrollers, platform, sensors.

Résumé

La technologie de l'Internet des Objets permet de connecter le monde physique au monde digital. Son architecture regroupe une partie matérielle (microcontrôleurs) et une partie logicielle (plateforme, application. . .). Le but de ce projet est de réaliser une plateforme qui permet de collecter des données issues d'un réseau de capteurs, mais qui permet également de stocker ces données pour d'éventuelles analyses ou études. Nous espérons, avec les résultats obtenus, contribuer à l'amélioration et la généralisation de cette technologie dans notre pays.

Mots-Clés : Internet des objets, microcontrôleurs, plateforme, capteurs.

ملخص

ملخص : تتيح تقنية إنترنت الأشياء توصيل العالم المادي بالعالم الرقمي. تشتمل هندسته على جزء من الأجهزة (متحكمات دقيقة) وجزء برمجي(موقع، تطبيق ، إلخ). الهدف من هذا المشروع هو إنشاء موقع يجمع البيانات من شبكة أجهزة استشعار ، ولكنه يخزن أيضًا هذه البيانات للتحاليل أو الدراسات المحتملة. نأمل مع النتائج التي تم الحصول عليها المساهمة في تحسين وتعميم هذه التكنولوجيا في بلدنا

الكلمات المفتاحية: انترنت الأشياء، متحكمات دقيقة، موقع، حساسات

Abréviations

Act/Cap	Actionneur/Capteur
BDD	Base de données
HTML	Hypertext markup language
CSS	Cascading style sheets
HTTP	Hypertext Transfer Protocol
ID	Identifier
UID	User Identifier
AID	Application Identifier
LID	Locals Identifier
IDE	Integrated Development Environment
IdO	Internet des Objets
IOT	Internet of things
IP	Internet Protocol
MCU	Micro-controller Unit
MQTT	Message Queuing Telemetry Transport
SQL	Sigle de Structured Query Language
Sub/pub	Subscriber/Publisher
TCP	Transmission Control Protocol
URL	Universal Serial Bus
Wifi	Wireless

Table des matières

Table des matières

Table des figures

Introduction générale

Depuis la dernière décennie, nous assistons à une révolution technologique dans le monde du digital. Au cœur de cette révolution, l'Internet des objets (IoT) joue un rôle majeur sur notre façon de vivre. Cette technologie, qui sera prochainement partie intégrante de notre quotidien, consiste à connecter le monde physique au monde digital. Tout ceci est aujourd'hui possible grâce à la connexion massive des objets physiques qui nous entourent. Ils deviennent plus autonomes et intelligents.

L'IoT consiste à doter nos objets physiques du quotidien de capteurs, actionneurs et d'unités de calcul leurs permettant de pouvoir réagir intelligemment dans des environnements complexes. Les données collectées sont ensuite échangées digitalement.

Aujourd'hui, les objets ne sont plus considérés comme des produits mais plutôt comme des services. On parle alors de maisons intelligentes où l'on peut contrôler les objets à distance, de montres intelligentes qui surveillent l'état de santé de l'utilisateur, etc.

L'Internet des Objets est une industrie riche qui pourrait changer radicalement l'équation économique du marché en apportant de nouveaux services et en créant de nouvelles opportunités.

L'Internet des Objets aspire à répondre aux exigences et besoins des utilisateurs, et c'est en développant de nouvelles plateformes et applications qu'il sera possible d'exploiter son potentiel immense.

Dans ce projet, nous allons nous intéresser à cette technologie dans le but de contribuer à son amélioration, et répondre à certaines exigences ou besoins. Nous allons donc réaliser une plateforme de gestion d'un réseau de capteurs intelligents, afin que les utilisateurs puissent contrôler et recevoir des données de leurs objets connectés.

Ce mémoire est composé de trois chapitres, le premier chapitre sera consacré à la présentation de cette technologie qui est l'Internet des Objets, nous allons également exposer ses domaines d'application, tout en citant ses avantages et inconvénients, et le matériel utilisé dans l'IOT.

Dans le deuxième chapitre, nous allons expliquer en détails la réalisation de notre plateforme, ainsi que la partie physique de notre architecture et le lien entre ces dernières.

Dans le troisième et dernier chapitre, nous allons exposer notre plateforme et présenter les résultats des expérimentations de la connectivité entre elle et les objets connectés de notre architecture.

Chapitre 1

Généralités sur les objets connectés

1.1 Introduction

L'Internet des objets (IOT) est un système qui consiste à connecter en réseaux différents types d'éléments électroniques, informatiques ou mécaniques [1]. Dans ce chapitre nous allons essayer d'aborder en détails les objets connectés et donner les points nécessaires pour comprendre comment fonctionne cette technologie.

FIGURE 1.1 – Schéma d'une architecture IOT

1.2 Aperçu historique sur l'Internet Des Objets

L'idée d'ajouter des capteurs et de l'intelligence à des objets physiques a été évoquée pour la première fois dans les années 1980 [2], lorsque des étudiants universitaires ont décidé de modifier un distributeur automatique de Coca-Cola pour suivre son contenu à distance. Mais la technologie était encombrante et les progrès étaient limités [3].

Le terme "Internet des objets" a été inventé en 1999 par l'informaticien Kevin Ashton. Alors qu'il travaillait chez Procter & Gamble, Ashton a proposé de placer des puces d'identification par radiofréquence (RFID) sur les produits afin de les suivre tout au long de la chaˆıne d'approvisionnement [4].

Pour attirer l'attention des dirigeants, il aurait intégré le mot "internet", alors très à la mode, dans sa proposition. Et la phrase est restée. Au cours de la décennie suivante, l'intérêt du public pour la technologie IoT a commencé à décoller, alors que de plus en plus d'appareils connectés arrivaient sur le marché. En 2000, LG a annoncé le premier réfrigérateur intelligent, en 2007 le premier iPhone a été lancé et en 2008, le nombre d'appareils connectés dépassait le nombre d'habitants de la planète.

En 2009, Google a commencé à tester des voitures sans conducteur et en 2011, le thermostat intelligent Nest de Google est arrivé sur le marché, permettant de contrôler à distance le chauffage central [5].

1.3 Les objets connectés

1.3.1 Définition

Comme nous l'avons mentionné dans l'introduction, les objets connectés peuvent également être appelés IoT, ou Internet des objets. L'IoT est un réseau géant d'objets connectés comprenant des téléphones portables, des écouteurs, des cafetières, des machines à laver, des lampes, des implants de moniteurs cardiaques, etc. Ces appareils IoT communiquent entre eux en transférant des données de réseaux communicables qui, à leur tour, connectent des objets physiques à l'internet, comme des applications [6].

L'utilisation d'objets connectés donne aux appareils le pouvoir de nous dire ce qui se passe dans un environnement sans y être physiquement. Cela implique le transfert de données sur un réseau sans nécessiter d'interaction entre humains ou entre humains et ordinateurs. Selon Google, l'IoT se divise en trois composantes : le dispositif, la passerelle et le nuage [7].

1.3.2 Dispositif

Il s'agit du matériel et du logiciel utilisés pour interagir avec le monde. Cela signifie que les appareils sont connectés à un réseau pour communiquer entre eux et avec des applications centralisées [8].

1.3.3 Passerelle

Le dispositif de passerelle IoT fournit une connexion et une traduction entre les appareils et le cloud. Il permet également aux appareils qui ne sont pas directement connectés à Internet d'utiliser les services du cloud. La passerelle traite les données de plusieurs appareils avant qu'elles ne soient toutes envoyées dans le cloud [9].

1.3.4 Nuage

Le cloud est essentiel pour stocker, traiter et analyser les big data. Les données de chaque appareil sont envoyées dans le cloud et traitées avec d'autres appareils. Parallèlement à ces 3 composants, les systèmes IoT sont composés d'actionneurs et de capteurs qui constituent l'infrastructure centrale d'un cadre IoT pour en assurer la précision. Ils permettent les interactions et la communication entre le monde physique et le monde numérique.

Les actionneurs sont des dispositifs utilisés pour manipuler l'environnement physique, comme le contrôle des vannes de température dans votre maison intelligente, et les capteurs collectent les données de l'environnement ou de l'objet surveillé, comme votre maison intelligente. Par exemple, un capteur peut reconnaître la température qui augmente rapidement dans votre maison, détectant ainsi la chaleur d'un incendie. Le capteur envoie ensuite ces données au centre de contrôle, et c'est là que l'actionneur entre en jeu en activant les sprinklers de votre maison pour éteindre le feu. Vous trouverez ci-dessous un schéma détaillé décrivant les rôles des actionneurs et des capteurs à l'aide de cet exemple [10].

1.4 Les avantages d'Internet Des Objets

Voici quelques avantages de l'IoT :

1.4.1 Communication

L'IoT encourage la communication entre les appareils, également connue sous le nom de communication Machine-to-Machine (M2M). Grâce à cela, les appareils physiques peuvent rester connectés, ce qui permet une transparence totale avec moins d'inefficacité et une meilleure qualité [11].

1.4.2 Automatisation et contrôle

Les objets physiques étant connectés et contrôlés numériquement et de manière centralisée grâce à l'infrastructure sans fil, le fonctionnement est largement automatisé et contrôlé. Sans intervention humaine, les machines sont capables de communiquer entre elles, ce qui permet d'obtenir des résultats plus rapides et opportuns.

1.4.3 Information

Il est évident que le fait de disposer de plus d'informations permet de prendre de meilleures décisions. Qu'il s'agisse de décisions banales comme le fait de savoir ce qu'il faut acheter à l'épicerie ou si votre entreprise a suffisamment de gadgets et de fournitures, la connaissance est le pouvoir et plus de connaissance est mieux.

1.4.4 Surveillez

Le deuxième avantage le plus évident de l'IoT est la surveillance. Connaître la quantité exacte de fournitures ou la qualité de l'air dans votre maison, peut fournir davantage d'informations qui n'auraient pas pu être collectées facilement auparavant. Par exemple, le fait de savoir que vous n'avez plus de lait ou d'encre d'imprimante peut vous éviter un autre voyage au magasin dans un avenir proche. En outre, le suivi de la date de péremption des produits peut et va améliorer la sécurité.

1.4.5 Temps

Comme le laissaient entendre les exemples précédents, le temps gagné grâce à l'IoT peut être considérable. Et dans la vie moderne d'aujourd'hui, nous avons tous besoin de plus de temps.

1.4.6 Argent

Le plus grand avantage de l'IoT est l'économie d'argent. Si le prix de l'équipement de marquage et de surveillance est inférieur à la somme d'argent économisée, alors l'internet des objets sera très largement adopté. L'IoT s'avère fondamentalement très utile pour les personnes dans leurs routines quotidiennes en permettant aux appareils de communiquer entre eux de manière efficace, ce qui permet d'économiser de l'énergie et de réduire les coûts. En permettant aux données d'être communiquées et partagées entre les appareils, puis en les traduisant en fonction de nos besoins, l'IoT rend nos systèmes efficaces [12].

1.4.7 L'automatisation des tâches quotidiennes

L'IoT permet d'automatiser et de contrôler les tâches quotidiennes, en évitant l'intervention humaine. La communication de machine à machine permet de maintenir la transparence des processus. Elle conduit également à une uniformisation des tâches. Elle permet également de maintenir la qualité du service. Nous pouvons également prendre les mesures nécessaires en cas d'urgence [13].

1.4.8 Efficacité et gain de temps

L'interaction machine-machine permet une meilleure efficacité, ce qui permet d'obtenir rapidement des résultats précis. Cela permet de gagner un temps précieux. Au lieu de répéter les mêmes tâches tous les jours, cela permet aux gens de faire d'autres travaux créatifs.

1.4.9 Économies d'argent

L'utilisation optimale de l'énergie et des ressources peut être obtenue en adoptant cette technologie et en gardant les appareils sous surveillance. Nous pouvons être alertés en cas d'éventuels goulets d'étranglement, de pannes et de dommages au système. Nous pouvons donc économiser de l'argent en utilisant cette technologie.

1.4.10 Une meilleure qualité de vie

Toutes les applications de cette technologie se traduisent par un confort accru, une plus grande commodité et une meilleure gestion, ce qui améliore la qualité de vie.

1.5 Objets physiques et virtuels

À l'heure actuelle, peu de systèmes traitent des éléments qui pourraient être considérés comme des objets virtuels, et il existe principalement deux approches différentes pour les modéliser [14].

1.5.1 Ressources gérées par des applications de dispositifs

Dans ce type de système, les objets virtuels sont composés d'un certain nombre d'enregistrements stockés dans une base de données. Les enregistrements dépendent d'une application qui gère l'application et contient la logique commerciale qui est exécutée par rapport aux enregistrements dans le magasin de données. Les informations associées à chaque enregistrement qui représente un objet virtuel varient en fonction de chaque système et de chaque logique métier . Il existe parfois des règles ou des conventions pour modéliser les objets virtuels appartenant à une logique métier spécifique, mais il n'existe pas de modèle général [15]. Les déficiences localisées de ces systèmes sont : L'un des objectifs est que les objets virtuels fonctionnent sur une grande variété de dispositifs. Les appareils qui veulent lire un objet virtuel doivent avoir installé une application spécifique qui reconnaît le format particulier de l'objet.

Dérivé du premier problème, les dispositifs nécessitent un grand nombre d'applications installées, si chaque application est seulement capable d'interpréter un

type particulier d'objet. En plus de la complication que représente le développement d'une application pour chaque type d'objet virtuel, de dispositif et de système d'exploitation. En suivant l'approche de l'internet des objets, les objets virtuels devraient également être capables de s'intégrer avec d'autres applications, dispositifs et utilisateurs. Lorsqu'un objet virtuel est conçu comme une source qui est traitée par une application spécifique, les difficultés sont les mêmes.

Les objets virtuels sont des objets auxquels une autre application ou un autre dispositif peut accéder ou qu'il peut découvrir de manière générique. Selon que l'application qui gère les mécanismes de l'objet virtuel offre ou non une certaine forme d'intégration, d'autres applications peuvent accéder à l'objet. Dans une situation idéale, un objet virtuel spécifique devrait être accessible à d'autres objets ou applications, et devrait être capable de faire de même. La communication avec les objets virtuels devrait se faire de manière standardisée, même s'il est évident que chaque objet ne fera pas les mêmes actions, mais la manière d'afficher et d'accéder à ces actions devrait être commune à tous.

1.5.2 Ressources gérées par des applications Web

Dans ces systèmes, les objets virtuels sont enregistrés dans un entrepôt de données et gérés par une application Web. Les utilisateurs interagissent avec les objets virtuels à l'aide d'un navigateur Web ; dans certains systèmes, ils disposent également de dispositifs spécifiques, tels que des distributeurs automatiques de billets (ATM), qui sont directement connectés à l'application de gestion. Cette approche fait que l'interprétation et la gestion de l'objet ne sont pas conditionnées par les applications installées sur le client, car elles se font à travers un navigateur web [16].

1.6 Composants de l'Internet des Objets

L'IoT n'est pas une technologie mais un système de systèmes. L'interopérabilité entre ces systèmes et l'intégration de tous les composants induisent une complexité forte. La capacité à gérer les interfaces est donc déterminante [17]. Lier un objet ou un lieu à Internet est un processus plus complexe que la liaison de deux pages Web. L'Internet des objets exige sept composants :

1. Une étiquette physique ou virtuelle pour identifier les objets et les lieux. Quelques systèmes d'étiquetage sont décrits ci-dessous. Pour permettre aux étiquettes physiques plus petites d'être localisées elles doivent être embarquées dans des marqueurs visuels.
2. Un moyen de lire les étiquettes physiques, ou de localiser les étiquettes virtuelles.
3. Un dispositif mobile tel qu'un téléphone cellulaire, un organiseur ou un ordinateur portable.
4. Un logiciel additionnel pour le dispositif mobile.
5. Un réseau sans fil de type 2G ou 3G afin de permettre la communication entre le dispositif portable et le serveur contenant l'information liée à l'objet étiqueté.
6. L'information sur chaque objet lié. Cette information peut être contenue dans les pages existantes de WWW, les bases de données comportant des informations de type prix, etc.

FIGURE 1.2 – Schéma de composants d'architecture IOT.

7. Un affichage pour regarder l'information sur l'objet lié. À l'heure actuelle, il est des plus probable que ce soit l'écran d'un téléphone mobile.

1.7 Caractéristiques fondamentales de l'Internet des Objets

1.7.1 Connectivité

L'une des caractéristiques les plus importantes dans le domaine de L'IOT est la connectivité, où tous les éléments peuvent être connectés sans une liaison directe par les ondes radio, Bluetooth, Wi-Fi, etc. Pour rendre la connectivité dans son état optimum en termes d'efficacité et de la surface de couverture en utilisant des protocoles de couches de connectivité internet comme on peut dans certains cas réaliser cette connectivité localement (intranet)

1.7.2 Détection

L'être humain, vit quotidiennement des expériences différentes lui permettent de comprendre et analyser toute situation rencontrée naturellement. L'IOT est basé sur la lecture d'un signal analogique, le convertir d'une manière permet d'extraire des informations significatives, en utilisant des différents types de capteurs tels que le GPS,

RFID, capteur de température, vitesse... afin de recueillir des données, des mesures ou des grandeurs selon l'étude qu'on est en train de faire [18].

1.7.3 Engagements actifs

L'engagement actif signifie la fonctionnalité et la connexion entre les éléments (produits, technologies et les services multiplateformes) via le dispositif IOT, on utilise souvent le cloud computing dans la blockchain afin de réaliser ces engagements entre les composants. Dans le domaine industriel, les données analogiques récupérées seront prétraitées et redimensionnées par rapport à la capacité de l'entreprise. Les statistiques indiquent que seulement la moitié de ces données sont structurées et 1% sont non structurées pour prendre des grandes décisions dans le domaine commercial. Le but de cet engagement est de réaliser un système capable de gérer des grandes quantités de données d'un grand réseau IOT, prenant en compte la possibilité que ce réseau sera plus vaste avec plus de quantité de données [19].

1.7.4 Échelle

La conception de l'IOT doit toujours être flexible aux besoins de l'utilisateur, elle dépend en termes d'échelle, où on peut la trouver dans les maisons intelligentes, les systèmes d'automatisation dans l'industrie aussi dans différents postes de travail.

1.7.5 Nature dynamique

L'IOT, est basé sur la collection et la conversion des données selon l'objectif voulu, ce processus consiste à rendre tout élément connecté (IOT) peut changer d'état d'une manière dynamique prenant comme exemple un capteur de température qui varie en fonction des notions de la météo, la localisation, etc.

1.7.6 Intelligence

Le développement du domaine IOT a connu l'intégration de l'intelligence artificielle. Pour avoir un système plus intelligent fluide et dynamique, des modèles d'apprentissage profond compatibles avec le réseau IOT sont réalisés, vu que ces deux technologies ont le but d'automatiser les taches d'un système et minimiser l'intervention humaine.

1.7.7 Énergie

L'écosystème IOT est un grand réseau IOT qui se compose de matériels, applications et logiciels connectés. Le fonctionnement de ce réseau nécessite beaucoup d'énergie, c'est pour cela qu'on cherche toujours à concevoir des systèmes où la consommation d'énergie soit minimale [20].

1.7.8 Sécurité

Les réseaux IOT peuvent avoir des vulnérabilités qui menacent le fonctionnement du système et même la confidentialité des entreprises ou des personnes. Pour éviter ces pertes d'informations, un système de sécurité doit être intégré lors de la conception du

l'IOT qui comprend chaque élément sans exception. Laisser un composant non sécurisé peut causer une défaillance du réseau IOT [21].

1.7.9 Intégration

L'internet des objets a l'atout d'intégrer plusieurs domaines au même temps ce qui rend l'utilité plus bénéfique pour l'utilisateur. Il assure également un bon compromis entre les coûts d'infrastructure et les coûts opérationnels.

1.8 Quels sont les composants d'une architecture IoT ?

L'architecture de l'IOT est composée essentiellement, de quatre éléments :

1.8.1 Composant d'application et d'analyse

C'est la pièce laquelle est responsable du traitement des infos récupérées du système IOT, puis les affichées. Elle contient plusieurs outils tel que l'intelligence artificielle, le deep learning, la capacité de visualisation, etc. comme elle des programmes et des logiciels professionnels d'application multifonctions comme IBM, SPSS, SAS. . . et aussi des outils et tableaux de bord IoT spécialisés des fournisseurs de cloud comme Google et Microsoft, ainsi que les applications fournies par des éditeurs de logiciels comme SAP et salesfoce [22].

1.8.2 Composant d'intégration

il assure l'intégration efficace de tous les composants qui forment l'architecture IOT avec les systèmes de gestions des entreprises comme le PGI (Progiciel de Gestion Intégré).

1.8.3 Composant de sécurité et de gestion

le réseau d'internet des objets doit toujours être en sécurité. La procédure de sécurisation se fait en deux niveaux, la sécurité des composants physiques et la sécurité intégrée en utilisants des fournisseurs comme Azure, LynxOS, Mocana. . . qui proposent des systèmes de sécurité pour l'IOT. On trouve aussi des packages de sécurité dédiés à l'IOT fournis par Forescout, Symantec et Trend Micro.

1.8.4 Composant d'infrastructure

Il s'agit des éléments physiques qui composent le réseau IOT des capteurs et des actionneurs qui font le traitement des informations et la gestion du réseau.

La relation entre les composants constitue la deuxième partie de l'architecture. Par relation entre, nous entendons la manière dont les composants communiquent entre eux et les types d'informations échangées. Il peut s'agir de flux de données, de flux de métadonnées, d'informations de contrôle ou d'aucune information. Les composants logiciels communiquent souvent via des API, et les composants de la couche réseau communiquent généralement via des protocoles réseau.

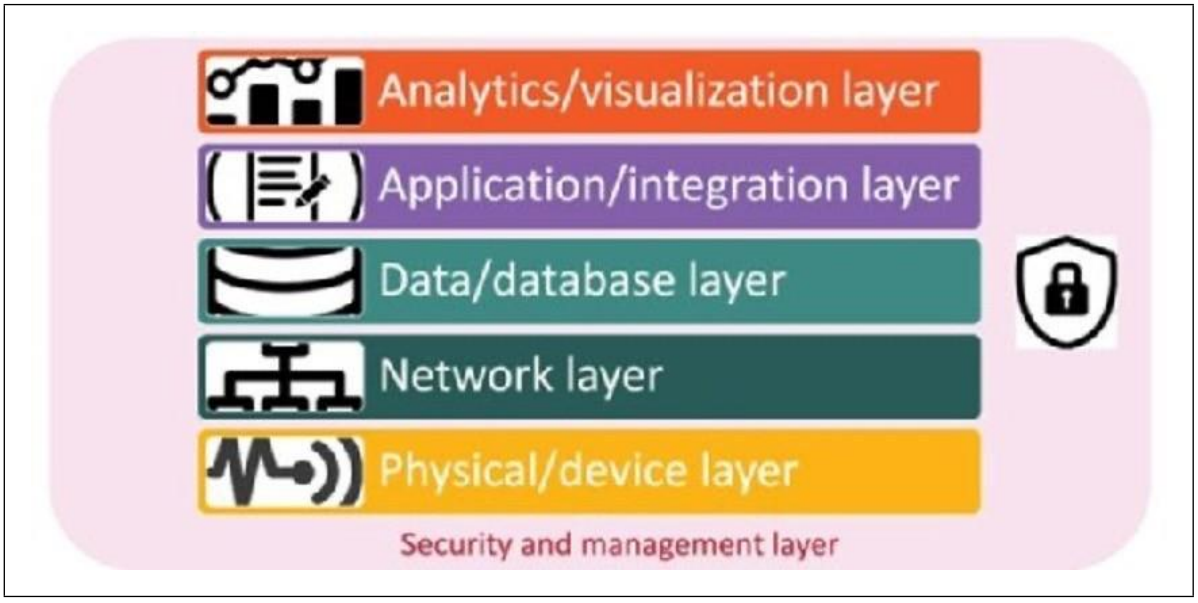


FIGURE 1.3 – Les couches d'une architecture IOT.

1.9 Les couches de l'architecture IOT

En parlant de couches, les architectes pensent souvent en termes de couches de composants, telles que la couche réseau, la couche perception, la couche traitement, la couche physique, la couche passerelle, la couche plate-forme, la couche dispositif, la couche métier, la couche sécurité, la couche capteur, etc [23]. Le concept d'une couche est qu'elle comprend un ensemble de capacités qui communiquent entre elles, mais pour les besoins d'autres composants peuvent être traités comme une seule entité, avec une seule entité transparente. Dans l'architecture IoT, la couche application n'a pas besoin de savoir quel type de réseau physique transporte les données. Tous les périphériques réseau constituent la couche réseau qui transporte le trafic selon les besoins des applications. Nous définissons les six couches de l'architecture IoT comme décrit ci-dessous. Notez que dans certains cas, les couches sont composées de sous-couches. Il s'agit d'une caractéristique commune aux architectures complexes, comme celle de l'IoT. (Voir la figure **??**).

1.9.1 couche physique/appareil

Il s'agit des capteurs, des actionneurs et des autres dispositifs intelligents et dispositifs connectés qui constituent la couche physique et la couche des dispositifs. Ces dispositifs intelligents capturent des données (capteurs), prennent des mesures (actionneurs) ou parfois les deux [24].

1.9.2 La couche réseau

Elle comprend les dispositifs de réseau et les types et protocoles de communication (5G, Wi-Fi, Bluetooth, etc.). Bien que de nombreuses architectures IoT reposent sur des couches réseau polyvalentes, la tendance est de plus en plus à l'adoption de réseaux dédiés à l'IoT.

1.9.3 La couche de données/base de données

Cette couche comprend également la couche de plateforme de base de données. Il existe toute une gamme de bases de données utilisées pour les architectures IoT, et de nombreuses organisations consacrent beaucoup de temps à la sélection et à l'architecture des bonnes bases de données IoT. Ensemble, la couche physique/appareil, la couche réseau et les couches données/base de données constituent le composant d'infrastructure évoqué ci-dessus.

1.9.4 Couche analyse/visualisation

Cette couche comprend la couche analytique, la couche de visualisation et la couche de perception. Essentiellement, cette couche se concentre sur l'analyse des données fournies par l'IoT et les met à la disposition des utilisateurs et des applications pour leur donner du sens.

1.9.5 La couche d'application/intégration

Il s'agit de la couche des applications et des plates-formes qui s'intègrent ensemble pour fournir la fonctionnalité de l'infrastructure IoT à l'entreprise. En d'autres termes, les couches d'application, de plateforme et d'intégration sont celles qui fournissent la valeur commerciale de l'infrastructure IoT. La couche de traitement et la couche métier font toutes parties de la couche d'application/intégration plus large.

1.9.6 Couche de sécurité et de gestion

Comme son nom l'indique, cette couche englobe à la fois la couche de sécurité et la couche de gestion. À proprement parler, il ne s'agit pas d'une couche, car elle est reliée à toutes les autres couches pour assurer la sécurité et la gestion. Mais c'est un composant important qui mérite d'être pris en compte à chaque couche. Ces couches vont de bas en haut d'une manière similaire au modèle Open Systems Interconnextion, qui est la source originale du concept de stratification. (Voir la figure 3 ci-dessous).

1.10 Domaines d'utilisation de l'IOT

1.10.1 L'IoD et le domaine médical

L'internet des objets est notamment largement utilisée dans le domaine de santé et ceux dans plusieurs appareils à savoir les machines à rayon X, les compteurs d'énergie, les moniteurs connectés et environs 90% des hôpitaux mondiaux, et ce dans le but d'augmenter la productivité de ces derniers et donc assurer une meilleure qualité des soins apportés aux malades [25]. Hormis ces appareils, l'IoD est utilisée au quotidien dans le suivi et la surveillance au niveau des établissements de santés ainsi que la maintenance, également dans les interventions chirurgicales et dans les différents contrôles à distance, et les différents services spécialisés dans la géolocalisation. La normalisation de l'utilisation de l'IoD dans le domaine médicale permet de découvrir de nouveaux modèles permettant aux employés d'avoir une meilleure productivité mais

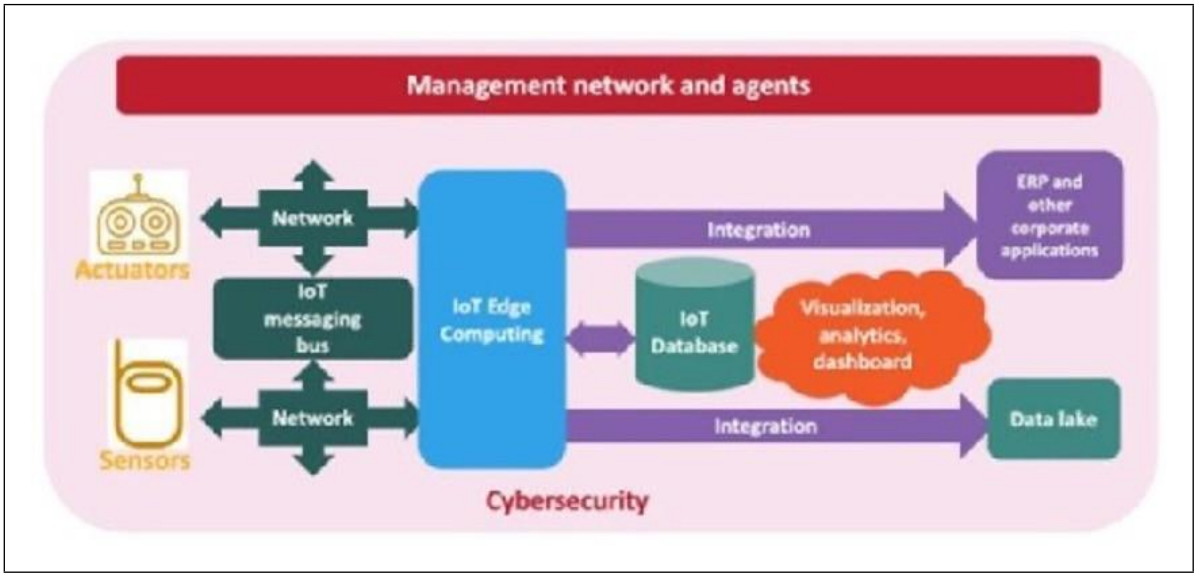


FIGURE 1.4 – Les couches d'une architecture IOT.

également une meilleure collaboration et communication entre les soignants entre eux et avec les patients.

1.10.2 Les maisons connectées ou les domotiques

Les maisons connectées ou les domotiques est un nouveau concept qui est en train de se normaliser ; une étude antérieure réalisée par le cabinet JUniper Research a prévu un accroissement de 200% dans le nombre des objets connectés au niveau des différents habitats en 2021 [26]. Outre les objets destinés au divertissement à savoir les télés intelligentes ou les smart-TVs ; les enceintes connectées, la domotique prend en considération d'autres domaines à savoir la sécurité, l'economie de la consommation d'energie au sein des domotiques ; et dans ce sens on distingue :

- Centrale domotique : Permet le contrôle et la programmation des differfenes intervenions ayant lieu au sein de l'habitat.
- Les capteurs d'informations : dont les alarmes de sécurité, la varation de températures de l'habitat ; etc.
- Actionneurs permettant de programmer et de contrôler même à distance plusieurs foyers.

1.10.3 L'industrie connectée

Le domaine industriel ; à l'image des autres domaines, utilisent également l'internet des objets profitant des bénéfices qu'elle lui apporte. L'utilisation des objets connectés est spécifique et répond à plusieurs besoins des industries :

- Besoin d'optimisation (concernant la chaine logistique).
- La transformation des processus d'entreprise.
- L'amélioration de l'efficacité et de la productivité.
- La traçabilité et la sécurité.

La révolution numérique offre également des opportunités pour certains types d'industries

de se renouveler et d'apporter de la valeur ajoutée à des domaines en perte de popularité. C'est le cas de SNCF Fret, par exemple, qui regagne doucement en crédibilité avec le lancement de ses locomotives connectées, qui permettent un meilleur suivi de ses wagons et une sécurité accrue pour les clients.

1.10.4 L'IoD et l'agriculture

La croissance rapide de la population mondiale, l'évolution des habitudes alimentaires et les dérèglements climatiques sont trois facteurs qui incluent les défis auxquels l'agriculture moderne est confrontée au quotidien. La productivité agricole doit augmenter de 70% d'ici 2050 pour répondre à la demande mondiale. Il ne s'agit pas seulement d'un défi technique, mais d'un problème humanitaire [27]. Les céréaliers et les maraîchers utilisent déjà des drones pour recueillir en temps réel des informations essentielles à la gestion des exploitations :

- Humidité du sol.
- Statut de plantation.
- Climat.. etc.

Les données collectées sont transmises à un tracteur connecté (parfois autonome). Cela permet une application fine des engrais et de l'arrosage sur des parcelles spécifiques et réduit les coûts financiers et énergétiques.

1.10.5 Le E-commerce

Les entreprises physiques subissent également des changements à l'ère numérique. Avec la concurrence féroce du e-commerce et du m-commerce, les magasins de détail cherchent à capitaliser sur la popularité de l'IoT en combinant le e-commerce avec le commerce de détail traditionnel. En conséquence, les magasins physiques se tournent également vers la révolution numérique, offrant de plus en plus de fonctionnalités amusantes et interactives pour améliorer l'expérience de vente et augmenter les conversions. Dans le concept de "smart retail", on retrouve la technologie d'identification par radiofréquence (RFID), qui améliore l'expérience client en proposant un parcours client hyper-personnalisé. En plus des applications mobiles, le concept de caddies connectés a été conçu pour faciliter les achats en supermarché [28].

- Liste de courses complète.
- Guider les itinéraires pour optimiser le temps d'exécution.
- Calcule automatiquement le nombre de paniers, etc.

Les commerçants investissent également dans des applications mobiles pour fidéliser et attirer les clients dans les magasins physiques. Par exemple, les promotions/ventes en cours sont notifiées lorsque les clients traversent le magasin.

1.10.6 Smart cities

Des villes dites «smart cities» ou encore «villes connectées» se sont engagées dans la transformation numérique pour répondre aux enjeux de la société moderne. L'urbanisation, l'économie et le développement durable sont des enjeux importants pour les villes du futur, qui doivent répondre aux exigences de populations de plus en plus denses et de ressources de plus en plus limitées [29]. En effet, la pollution et la surpopulation sont deux des principaux problèmes des agglomérations, qui doivent faire

face aux défis économiques et sociaux d'une population croissante sans dégrader la qualité de l'environnement dans lequel elles sont implantées. Encore une fois, l'IoT est utile :

— Automatisation de la maison.
— Médias numériques.
— Capteurs et compteurs intelligents.
— Borne de recharge pour véhicule électrique.
— Éclairage urbain intelligent, etc.

Les applications des systèmes en réseau sont nombreuses et constituent un véritable vivier d'innovation pour les professionnels de l'informatique et du numérique. Dans les villes intelligentes, la gestion de la consommation individuelle est mieux maîtrisée, ainsi que le traitement de la consommation urbaine à grande échelle (coûts et pollution des transports publics et urbains, élimination des déchets, gestion de l'eau, de l'électricité, etc.).

1.10.7 Internet des objets pour la sécurité routière

Les voitures connectées sont devenues particulièrement populaires ces dernières années et jouent un rôle important dans l'amélioration de la sécurité routière. La révolution numérique a ouvert des perspectives sans précédent pour l'industrie automobile comme la boîte d'appel d'urgence autonome , le tableau de bord synchronisé avec smartphone et le développement d'applications sur une plateforme dédiée... Les voitures d'aujourd'hui deviennent de véritables ordinateurs, menant lentement aux voitures autonomes actuellement testées chez Google. Les systèmes qui automatisent certaines tâches de conduite (allumage des phares, aide au stationnement, freinage automatique) rendront les véhicules de plus en plus autonomes, même s'ils ne savent pas encore conduire eux-mêmes [30]

1.11 Les défis de l'Internet des objets (IoT)

Hormis son développement rapide lui permettant d'occuper une grande partie dans la vie des êtres humains et dans la façon dont ils vivent, communiquent et font leurs affaires, l'internet des objets fait toujours face à de nombreux defis, de tous types auxquelles on va s'intéresser.

1.11.1 Défis de sécurité dans l'IoT

1.11.1.1 Absence de cryptage

Le cryptage est l'un des meilleurs moyens pour faire face aux pirates en leurs bloquant l'acces aux différentes données des utilisateurs. Il est également l'un des défis de sécurité principaux que l'IoT rencontre, et ceci est dû au faite que les disques ont des capacités de stockage et de traitement importante qu'on trouve sur un oridnateur ordinaire, ce qui augmente les attaques et donc les pirates auront facilement la capacité de manipuler les différents algorithmes conçues specialement pour la protection [31].

1.11.1.2 Tests et mises à jour insuffisants

L'augmentation importante des appareils IoT a obligé les producteurs de produire plus rapidement leur appareil et de les livrer dans des délais assez courts, et donc par conséquents, moins d'importance est accordée a leur sécurité. La majorité des appareils IoT fournies n'ont pas été suffisamment testes ni mis à jour et sont sujets aux attaques des pirates et à d'innombrables problèmes de sécurités.

1.11.1.3 Brute forcing et risque de mots de passe par défaut

Des informations d'identification et des informations d'identification faibles rendent presque tous les appareils IoT vulnérables au craquage de mot de passe et à la force brute. Les organisations qui utilisent des informations d'identification par défaut sur les appareils exposent leur entreprise et leurs actifs, leurs clients et leurs précieuses informations aux attaques par force brute.

1.11.1.4 IoT Malware et ransomware

Augmente avec le nombre d'appareils. Les rançongiciels utilisent le cryptage pour cibler efficacement les utilisateurs sur tous les appareils et plates-formes, tout en continuant à utiliser les précieuses données et informations des utilisateurs. Exemple -Les pirates peuvent détourner les caméras des ordinateurs et prendre des photos. En utilisant un point d'accès malveillant, les pirates peuvent exiger une rançon pour déverrouiller l'appareil et restituer les données.

1.11.1.5 La durée de vie de la batterie est une limitation

Problèmes d'emballage et d'intégration de petites puces avec un poids léger et une faible consommation d'énergie. Si vous avez suivi l'espace mobile, vous avez probablement vu que chaque année, il semble n'y avoir aucune limite en matière de taille d'affichage. Prenez l'essor des "phablettes", qui peuvent être des téléphones presque aussi gros que des tablettes. Bien qu'utile, un écran plus grand n'est pas toujours pratique, mais la taille de l'écran augmente pour accueillir une batterie plus grande. Les ordinateurs sont devenus de plus en plus minces, mais la puissance de la batterie est restée la même.

1.11.1.6 Augmentation des coûts et des délais de mise sur le marché

Les systèmes embarqués sont quelque peu limités par le coût. De meilleures méthodes sont nécessaires pour gérer la modélisation des coûts ou le coût optimal des composants électroniques numériques lors de la conception de dispositifs IoT.Les concepteurs doivent également résoudre les problèmes de temps de conception et commercialiser les dispositifs embarqués au bon moment.

1.11.1.7 Sécurité du système

La conception et la mise en œuvre du système doivent être robustes et fiables, et doivent être protégées à l'aide d'algorithmes cryptographiques et de procédures de sécurité. Cela implique différentes approches pour sécuriser tous les composants d'un système embarqué, du prototype au déploiement.

1.11.1.8 Capacité multiplateforme

Les applications IoT doivent être développées en gardant à l'esprit les futurs changements technologiques. Son développement nécessite un équilibre entre les capacités matérielles et logicielles. Le défi pour les développeurs d'applications IoT est de s'assurer que les appareils et plates-formes IoT produisent des performances optimales à un moment où les taux d'équipement et les réparations sont élevés.

1.11.1.9 Collecte et traitement des données

Les données joueront un rôle important dans le développement de l'IoT. Le facteur décisif ici est le traitement ou la facilité d'utilisation des données stockées. Outre la sécurité et la confidentialité, les équipes de développement doivent planifier correctement la manière dont les données sont collectées, stockées ou traitées dans l'environnement.

1.12 Sécurité dans l'Internet des Objets

Les applications IoT étendent les capacités des objets intelligents, mais elles peuvent également introduire des failles de sécurité. Risques et défis de sécurité liés à l'Internet des objets L'Internet des objets est plus ouvert et inclusif que le Machine-to-Machine (M2M) traditionnel et utilise différents types de réseaux, y compris Internet. Par conséquent, le système devient plus visible et potentiellement plus vulnérable. Par exemple, fin 2020, une machine à laver en réseau non sécurisée par défaut a été piratée pour accéder aux systèmes informatiques d'un établissement de santé et à toutes les données des patients. Si l'objet lui-même n'est pas un appareil sensible, il peut donner accès à des informations sensibles. Cette technologie, et le fait qu'elle soit connectée au réseau IoT de l'hôpital, la rend aussi vulnérable que n'importe quel ordinateur ou serveur et doit être protégée en conséquence. De nombreux appareils connectés manquent de fonctionnalités de sécurité. Cela est dû à l'hypothèse (erronée) selon laquelle les données collectées ne violent pas la vie privée. Il est donc demandé à l'utilisateur de protéger l'objet avec un simple mot de passe. Cependant, toutes les informations, applications et connexions dont les objets intelligents ont besoin pour fonctionner peuvent être exploitées par des individus malveillants. La protection des données collectées est un enjeu majeur de l'IoT, comme la protection des données personnelles dans les formulaires web. D'autres problèmes incluent le piratage d'objets intelligents et les menaces de logiciels malveillants. Les appareils IoT peuvent être attaqués par des pirates, des pirates. Les objets connectés avec des niveaux de sécurité faibles ou mal configurés par les utilisateurs peuvent être plus attractifs à des fins malveillantes [32].

1.13 Des solutions pour protéger les objets connectés

Alors que les menaces et les conséquences que le monde IoT peut créer sont reconnues, le concept de sécurité pour l'Internet des objets devient de plus en plus une condition préalable à son développement. Des mesures et normes de sécurité sont progressivement mises en place pour limiter la conception et l'utilisation de ces objets connectés afin de réduire les principales vulnérabilités de l'appareil et les risques de piratage. Pour protéger

tous les appareils dans tous les domaines de l'IoT, plusieurs couches de sécurité doivent être envisagées et améliorées [33].

- Capteur : Ce sont des mécanismes pour recueillir les données que vous recherchez. Il existe plusieurs façons de détecter rapidement les attaques à distance ou le vol tel que la détection des changements d'IMEI du réseau mobile, suppression des mouvements inattendus, surveillance du nombre de connexions, etc.
- Les réseaux, les flux de données et leur transport : Divers réseaux utilisés dans les projets IoT ont déjà mis en place des protocoles de sécurité solides pour assurer la sécurité de votre organisation.
- Stockage des données : Les plateformes informatiques doivent être sécurisées et les interventions humaines et techniques surveillées.
- Chaîne complète de l'objet à l'application : Les objets doivent être sécurisés et les flux de communication chiffrés de bout en bout.

1.14 Conclusion

Ce chapitre est réalisé pour donner une vue globale sur le domaine d'Internet des Objets. On a présenté quelques notions sur l'IOT, où on a défini l'Internet des objets et ses composants, on a parlé aussi du développement de ce domaine depuis son existence avec ses avantages et ses inconvénients, ainsi que son principe de fonctionnement et les défis rencontrés dans des différents domaines. L'évolution de ce domaine a touché une grande partie de nos activités quotidiennes et professionnelles de façon que notre vie devient de plus en plus facile et dynamique.

Chapitre 2

Conception et fonctionnement de l'architecture

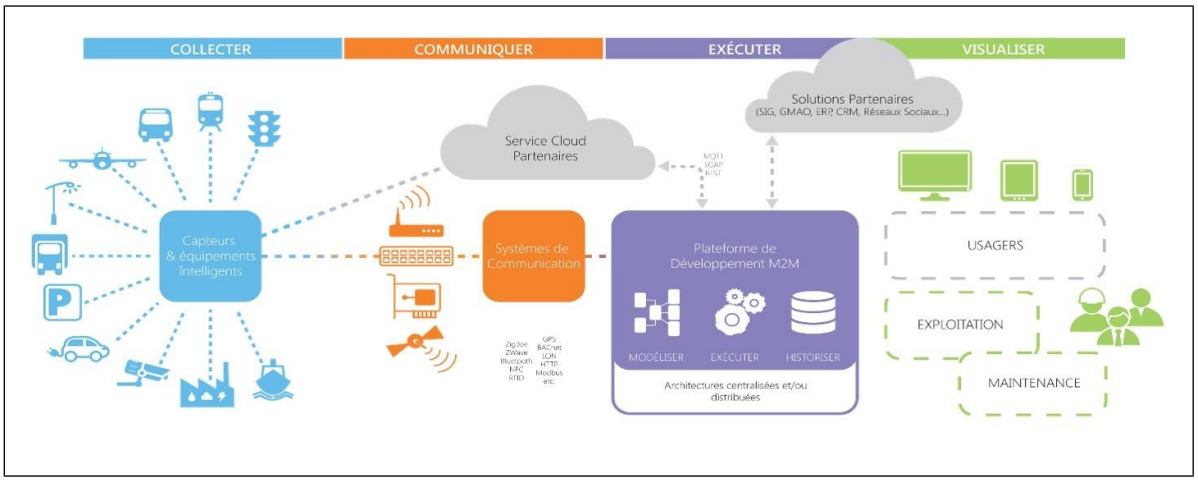


FIGURE 2.1 – Architecture physique de l'Internet des Objets.(https ://blog.engineering.publicissapient.fr/)

2.1 Introduction

L'Internet des objets fait référence aux appareils physiques qui reçoivent et transmettent des données sur des réseaux sans fil, avec une intervention humaine limitée. Cette technologie repose sur l'intégration d'un système informatique.Nous aurons donc une partie matérielle et une partie logicielle. Dans notre projet, la partie matérielle regroupe des cartes à microcontrôleurs (ESP8266) et les capteurs, quant à la partie logicielle elle regroupe les langages de programmation, la plateforme, et le protocole MQTT qui est responsable du transfert des données dans notre architecture. Dans ce chapitre, nous allons expliquer en détails toutes les étapes de la réalisation de notre plateforme, ainsi que la partie matérielle qui concerne la carte ESP8266 et les capteurs.

2.2 Architecture IOT

Le but de notre projet est de créer une plateforme pour la gestion d'un réseau de capteurs intelligents (ou objets connectés). Notre architecture est donc composée de trois parties : Une partie Software (ou logicielle) : Elle représente notre plateforme et ses différentes applications permettant au client de gérer son réseau. Une partie Hardware (ou matérielle) : Elle représente nos capteurs ou objets connectés. Une partie pour la Communication : Elle représente le protocole qu'on a utilisé pour permettre la connexion entre nos capteurs et notre plateforme

L'architecture illustrée ci-dessus, représente le schéma physique de l'Internet des Objets, nous expliquons par la suite les différentes parties de cette architecture.

2.2.1 Capteur et Actionneur

Dans cette partie, on parle d'objet connecté (ou de capteur intelligent), qui permet de relever des mesures de l'environnement, comme la température à titre d'exemple. On a aussi le cas d'un actionneur qui permet d'agir sur l'environnement.

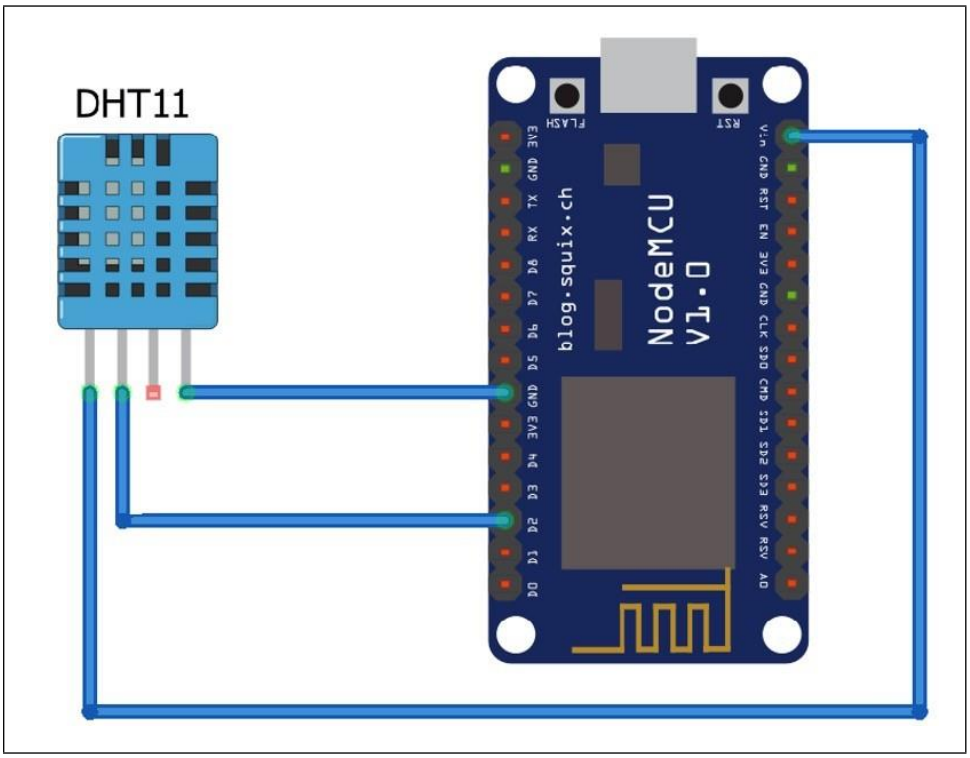


FIGURE 2.2 – Schéma de branchement du capteur avec le microcontrôleur. (www.iodesignpro.com).

2.2.2 Communication

Dans cette partie, les mesures prélevées seront transférées vers un broker mqtt (mosquitto). Nous utilisons une carte ESP8266 dotée d'une connexion wifi. La communication est assurée grace au protocole MQTT. Ces données seront par la suite collectées par la plateforme (la base de données plus précisément) à l'aide du client Paho MQTT et un programme Python.

2.2.3 Exécution

La plateforme se charge de la collecte, du traitement et du stockage des données, mais également de contrôler les objets en question.

2.2.4 Visualisation

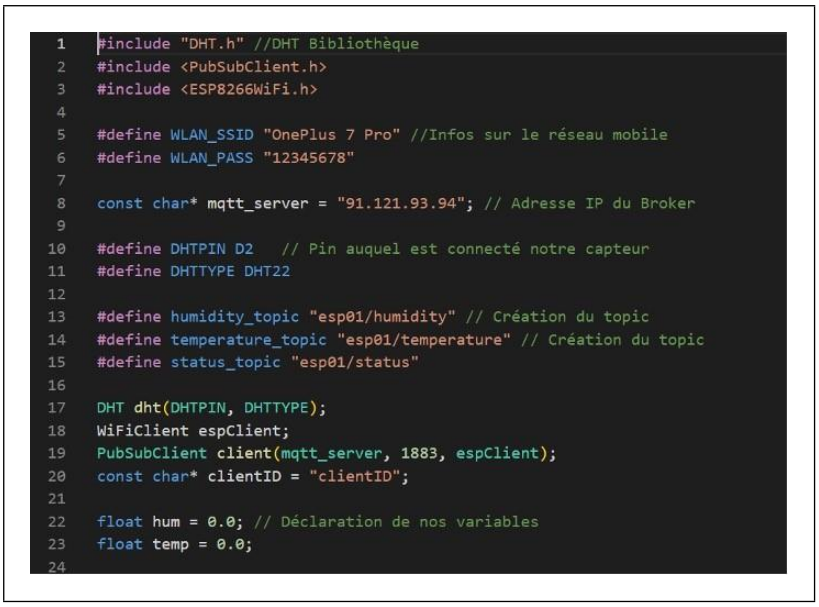
Dans la dernière partie, les données collectées seront affichées ou présentées à l'utilisateur sur une page de la plateforme. Ce dernier pourra donc superviser son réseau mais également constater les changements subis sur l'environnement par les actionneurs.

2.3 Réalisation de la partie physique

Dans cette partie, on a un capteur de température/humidité de type DHT22, connecté à un microcontrôleur ESP8266. Le schéma ci-dessous illustre le branchement de notre capteur avec le microcontrôleur :

La PIN du milieu a pour rôle de transférer les mesures, et les deux autres sont des

PIN d'alimentation (Une branchée avec la terre GND, et l'autre avec l'alimentation Vin). Après branchement, on doit écrire un programme dans l'IDE Arduino afin de configurer la connexion du module au réseau internet, mais aussi de programmer le transfert des données collectées par le capteur vers le serveur courtier (Broker, Mosquitto dans notre cas). Ci-dessous la programme Arduino en question :



```
#include "DHT.h" //DHT Bibliothèque
#include <PubSubClient.h>
#include <ESP8266WiFi.h>

#define WLAN_SSID "OnePlus 7 Pro" //Infos sur le réseau mobile
#define WLAN_PASS "12345678"

const char* mqtt_server = "91.121.93.94"; // Adresse IP du Broker

#define DHTPIN D2   // Pin auquel est connecté notre capteur
#define DHTTYPE DHT22

#define humidity_topic "esp01/humidity" // Création du topic
#define temperature_topic "esp01/temperature" // Création du topic
#define status_topic "esp01/status"

DHT dht(DHTPIN, DHTTYPE);
WiFiClient espClient;
PubSubClient client(mqtt_server, 1883, espClient);
const char* clientID = "clientID";

float hum = 0.0; // Déclaration de nos variables
float temp = 0.0;

```

FIGURE 2.3 – Programme d'envoie de données entre la carte ESP et le Broker 1

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  setup_wifi();  // Connexion à Internet
  client.setServer(mqtt_server, 1883);
  if (!client.connected()) {
    reconnect();
  }
}

void loop() {

  delay(2000);

  client.loop();

  hum = dht.readHumidity(); // Attribution des valeurs du capteur à la variable
  temp = dht.readTemperature();

  Serial.print(F("Humidity: "));
  Serial.print(String(hum).c_str());
  Serial.println(" %");
  client.publish(humidity_topic, String(hum).c_str()); // Publication sur le topic

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" Grad Celsius");
  client.publish(temperature_topic, String(temp).c_str(), true);

  delay(300000);
}
```

FIGURE 2.4 – Programme d'envoie de données entre la carte ESP et le Broker 2.

```
57  void setup_wifi() {
58    delay(10);
59    Serial.println();
60    Serial.print("Connecting to ");
61    Serial.println(WLAN_SSID);
62    int _try = 0;
63    WiFi.begin(WLAN_SSID, WLAN_PASS);
64    while (WiFi.status() != WL_CONNECTED) {
65     Serial.print(".");
66     delay(500);
67    }
68    Serial.println("");
69    Serial.println("WiFi connected");
70    Serial.println("IP address: ");
71    Serial.println(WiFi.localIP());
72   }
73
74  void reconnect() {
75    while (!client.connected()) {
76      Serial.print("Attempting MQTT connection...");
77      if (client.connect(clientID)) {
78        Serial.println("connected");
79        client.publish(humidity_topic, String(hum).c_str(), true);
80      } else {
81        Serial.print("failed, state: ");
82        Serial.println(client.state());
83        Serial.println("attempting reconnection...");
84        delay(300000);
85      }
86    }
87  }
```

FIGURE 2.5 – Programme d'envoie de données entre la carte ESP et le Broker 3.

En premier lieu, on établit une connexion à Internet, ce qui nous permettra de transférer les données vers le courtier (Broker) via Wifi. En second lieu, on se connecte au courtier, en introduisant l'adresse IP de ce dernier et le port que l'on choisit pour le transfert des données. En dernier lieu, on attribue les valeurs mesurées aux variables qu'on a créées, et on les envoie au courtier.

2.4 Réalisation de la plateforme

2.4.1 Les langages de programmation utilisés

2.4.1.1 Les langages de développement web

HTML : HTML est l'abréviation de HyperText Markup Language. Il s'agit d'un langage de balisage standard pour la création de pages Web. Il permet de créer et de structurer des sections, des paragraphes et des liens à l'aide d'éléments HTML (les éléments constitutifs d'une page Web) tels que les balises et les attributs. Il est également à noter que le HTML n'est pas considéré comme un langage de programmation, car il ne peut pas créer de fonctionnalités dynamiques. Il est désormais considéré comme une norme officielle du Web. Le World Wide Web Consortium (W3C) maintient et développe les spécifications du HTML, tout en fournissant des mises à jour

régulières.

CSS :CSS est l'acronyme de Cascading Style Sheets et est utilisé pour styliser les éléments écrits dans un langage de balisage tel que le HTML. Il sépare le contenu de la représentation visuelle du site. La relation entre le HTML et le CSS est fortement liée puisque le HTML est la base même d'un site et que le CSS représente toute l'esthétique d'un site Web.

JavaScript : JavaScript est un langage de programmation léger que les développeurs Web utilisent couramment pour créer des interactions plus dynamiques lors du développement de pages Web, d'applications, de serveurs, voire de jeux. Les développeurs utilisent généralement JavaScript parallèlement à HTML et CSS. Ce langage de script fonctionne bien avec CSS pour formater les éléments HTML. Cependant, JavaScript maintient toujours l'interaction avec l'utilisateur, ce que CSS ne peut faire seul. Il crée des éléments destinés à améliorer l'interaction des visiteurs avec les pages Web, tels que des menus déroulants, des graphiques animés et des couleurs de fond dynamiques.

2.5 Le Langage Python

Python est un langage de programmation interprété, orienté objet, et de haut niveau. Ses structures de données intégrées, combinées au typage dynamique et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de colle pour connecter des composants existants. La syntaxe simple et facile à apprendre de Python privilégie la lisibilité et réduit donc le coût de la maintenance des programmes. Python supporte les modules et les packages, ce qui encourage la modularité des programmes et la réutilisation du code. L'interpréteur Python et la bibliothèque standard étendue sont disponibles gratuitement sous forme de source ou de binaire pour toutes les principales plates-formes et peuvent être distribués librement.

2.5.1 Avantages et inconvénients du langage python

On commence par les avantages :

- Facile à lire, à apprendre et à écrire : La syntaxe de Python ressemble à celle de l'anglais. Il est donc plus facile de lire et de comprendre le code.
- Langage interprété : Ce qui signifie que Python exécute directement le code ligne par ligne. En cas d'erreur, il arrête l'exécution et signale l'erreur rencontrée.
- Typage dynamique : Python ne connaît pas le type de la variable avant l'exécution du code. Il attribue automatiquement le type de données pendant l'exécution. Le développeur n'a donc pas à se soucier de la déclaration des variables et de leurs types.
- Gratuit et open-source : Python est fourni sous la licence open-source approuvée par l'OSI (Open Source Initiative), il peut donc être utilisé et distribué gratuitement.
- Support de vastes bibliothèques : La bibliothèque standard de Python est énorme, on y trouve presque toutes les fonctions nécessaires. On n'a donc pas besoin de dépendre de bibliothèques externes. Mais même en cas de besoin, un gestionnaire de paquets Python (pip) facilite l'importation d'autres paquets.

— Portabilité : Dans de nombreux langages comme C/C++, on doit modifier notre code pour exécuter le programme sur différentes plateformes. Ce n'est pas le cas avec Python, On n'écrit le code qu'une seule fois qu'une fois et on l'exécute partout.

Et pour les inconvénients :

— Faible vitesse : L'exécution ligne par ligne du code conduit souvent à une exécution lente.
— Pas efficace en mémoire : Pour offrir la simplicité au développeur, Python doit faire un petit compromis, autrement dit utiliser une grande quantité de mémoire. Cela peut être un inconvénient lors de la création d'applications, alors que l'on préfère l'optimisation de la mémoire.
— Faible dans ledéveloppement mobile : Python est généralement utilisé pour la programmation côté serveur. On n'a pas l'occasion de voir Python du côté client ou dans les applications mobiles.
— Erreurs d'exécution : Comme expliqué ci-dessus, Python est un langage à typage dynamique. Le type de données d'une variable peut donc changer à tout moment. Une variable contenant un nombre entier peut contenir une chaîne de caractères dans le futur, ce qui peut entraîner des erreurs d'exécution. Par conséquent, les développeurs doivent effectuer des tests approfondis de leurs applications.

2.6 Le framework Django

Un framework en programmation est un outil qui fournit des composants prêts à l'emploi ou des solutions personnalisées afin d'accélérer le développement, le code personnalisé fait appel à des bibliothèques pour accéder à du code réutilisable.

Django est un framework web Python de haut niveau qui permet de développer rapidement des sites web sécurisés et faciles à maintenir. Conçu par des développeurs expérimentés, il se charge d'une grande partie des tâches fastidieuses liées au développement Web, ce qui permet de se concentrer sur l'écriture de l'application sans avoir à réinventer la roue. Il est gratuit et open source, dispose d'une communauté active, d'une excellente documentation et de nombreuses options d'assistance gratuite et payante.

Django vous aide à créer des applications web qui sont : complètes, polyvalentes, sûres, évolutives, faciles à maintenir et portables. La figure ci-dessous représente les principaux éléments de Django :

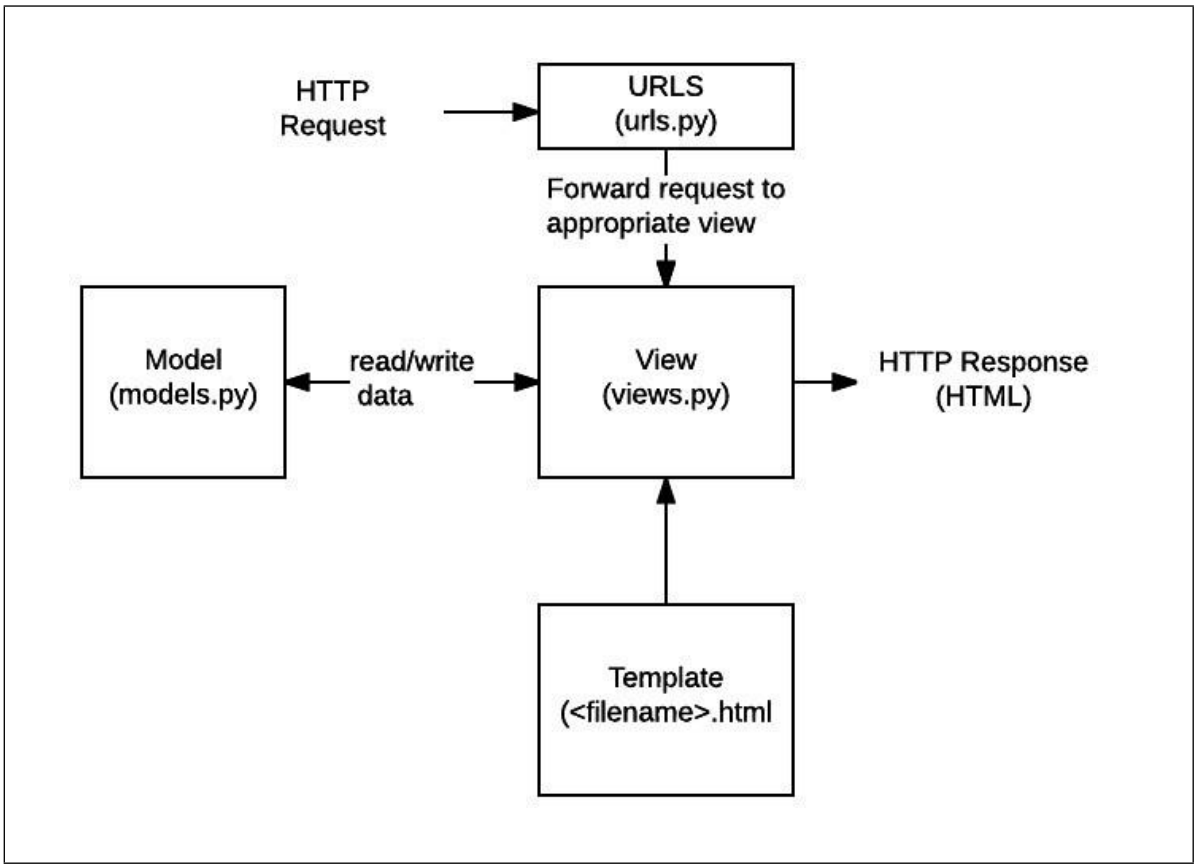


FIGURE 2.6 – Les éléments principaux de Django.

URLs : Bien qu'il soit possible de traiter les demandes provenant de chaque URL via une seule fonction, il est beaucoup plus facile à maintenir d'écrire une fonction de vue distincte pour traiter chaque ressource. Un mappeur d'URL est utilisé pour rediriger les demandes HTTP vers la vue appropriée en fonction de l'URL de la demande.

Vue : Une vue est une fonction de traitement des demandes, qui reçoit les demandes HTTP et renvoie les réponses HTTP. Les vues accèdent aux données nécessaires pour satisfaire les demandes via des modèles.

Modèles : Les modèles sont des objets Python qui définissent la structure des données d'une application et fournissent des mécanismes pour gérer (ajouter, modifier, supprimer) et interroger les enregistrements dans la base de données.

Templates : Un Template est un fichier texte définissant la structure ou la mise en page d'un fichier (tel qu'une page HTML), avec des espaces réservés utilisés pour représenter le contenu réel. Une vue peut créer dynamiquement une page HTML à l'aide d'un modèle HTML, en la remplissant de données provenant d'un modèle. Un modèle peut être utilisé pour définir la structure de n'importe quel type de fichier.

2.7 La réalisation de la plateforme

2.7.1 L'installation du framework Django

Comme première étape on installe le framework Django à travers la commande pip install django sur Windows powershell ou la fenêtre cmd :

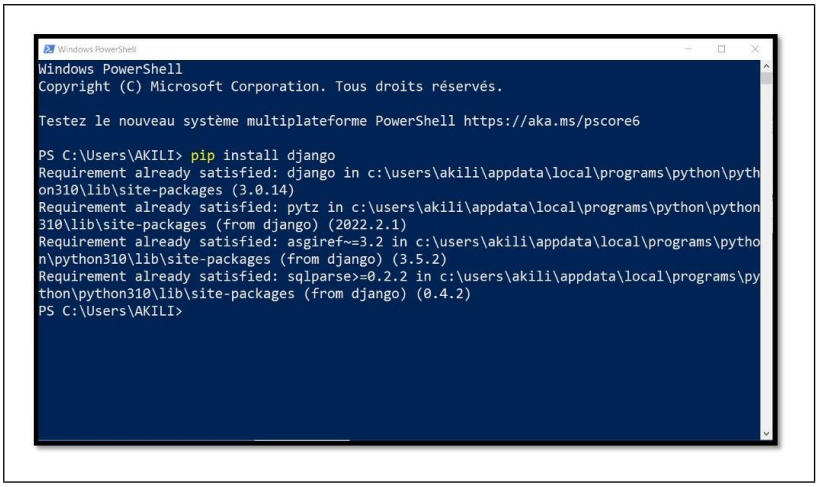


FIGURE 2.7 – Cammande d'installasion du framework Django.

On peut vérifier la version du Django installé en exécutant les commandes suivantes :

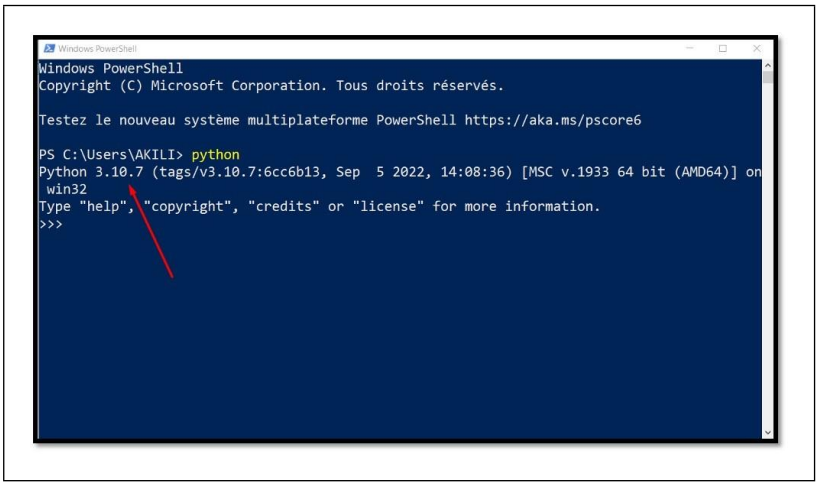


FIGURE 2.8 – La version Django utilisée.

2.7.2 La création du projet Django

La plateforme est une application du projet Django représenté comme un répertoire sur un emplacement de notre choix .Django nous permet de le créer en exécutant la commande django-admin startprojetct + le nom du projet ''myWeb" :

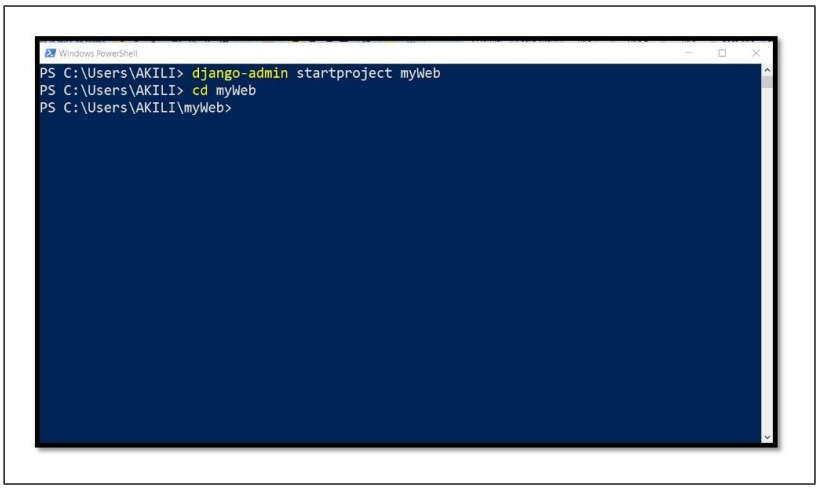


FIGURE 2.9 – Cammande de changement du chemin du projet Django.

Remarque : la commande cd (change directory) nous permet d'aller au fichier voulu.

On vérifie le répertoire dans le chemin indiqué qui contient un fichier manage.py qui permet de gérer le projet Django à travers la commande cmd

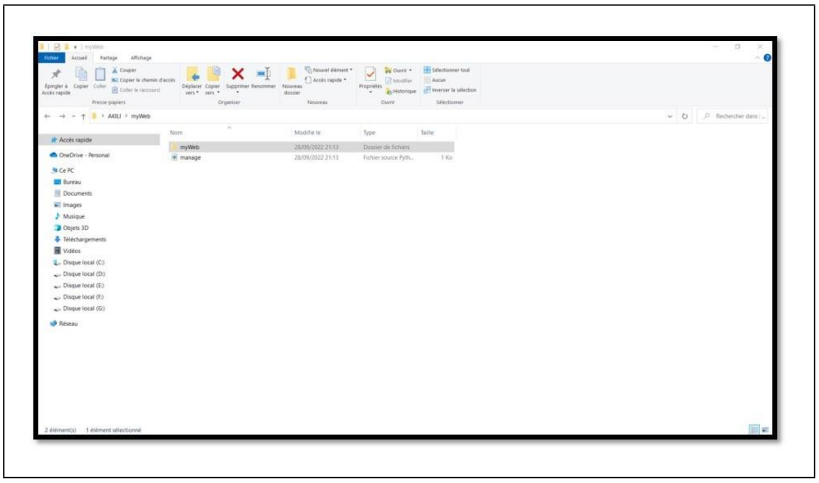


FIGURE 2.10 – La création du fichier du projet Django.

Il contient aussi d'autres fichiers python qui permettent de créer des applications, les gérer, les afficher ou faire l'interconnexion avec des serveurs, des bases de données, etc.

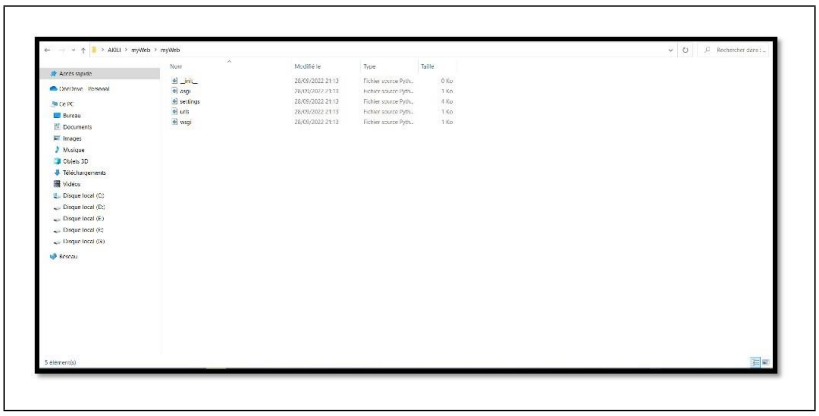


FIGURE 2.11 – Les fichiers Python que le projet Django les contient.

Pour visualiser le projet sur un navigateur, on tape la commande python manage.py runserver , le projet va s'afficher sur l'adresse locale 127.0.0.1 par le port 8000 par défaut :

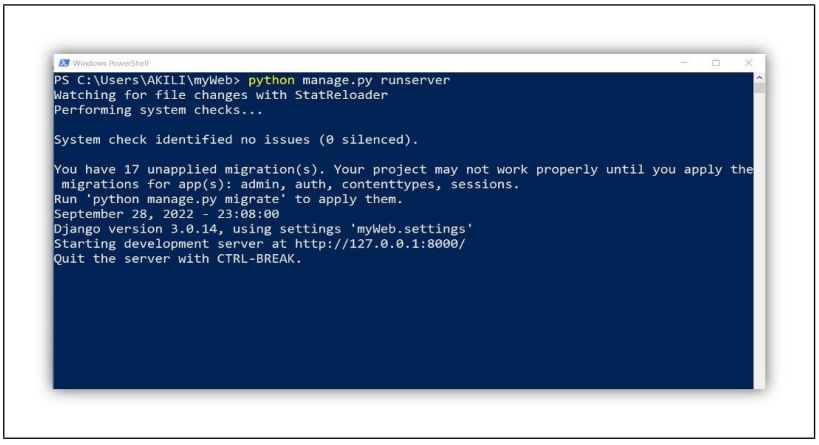


FIGURE 2.12 – La cammande du lancement du serveur Django.

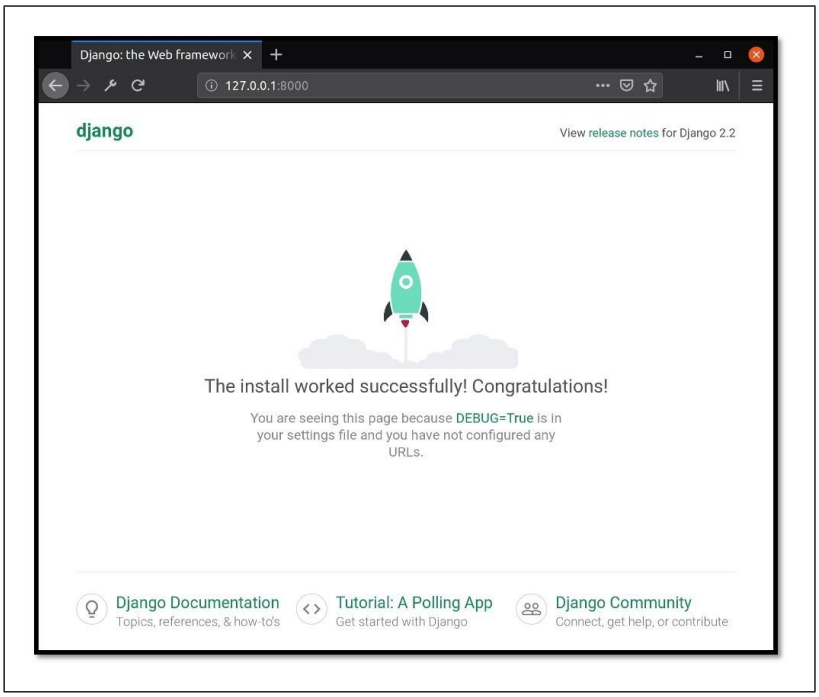


FIGURE 2.13 – La platform Django.

Remarque : pour changer le port on exécute la commande **python manage.py runserver + le port voulu**

2.7.3 La création d'un administrateur pour notre projet Django

Le lancement du serveur pour connecter le projet Django va nous créer automatiquement une base de données de type sqlite3,

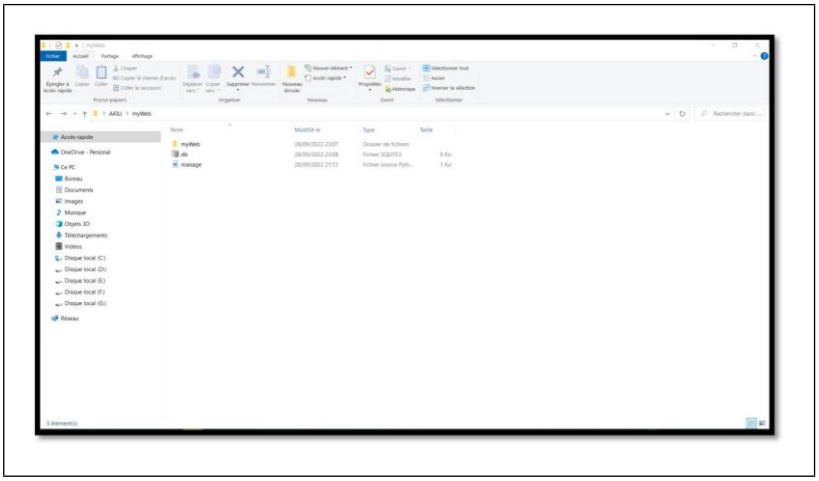


FIGURE 2.14 – La base de données SQLLITE3

On utilise le DB browser pour l'ouvrir et on aura une base de données qui ne contient aucune table :

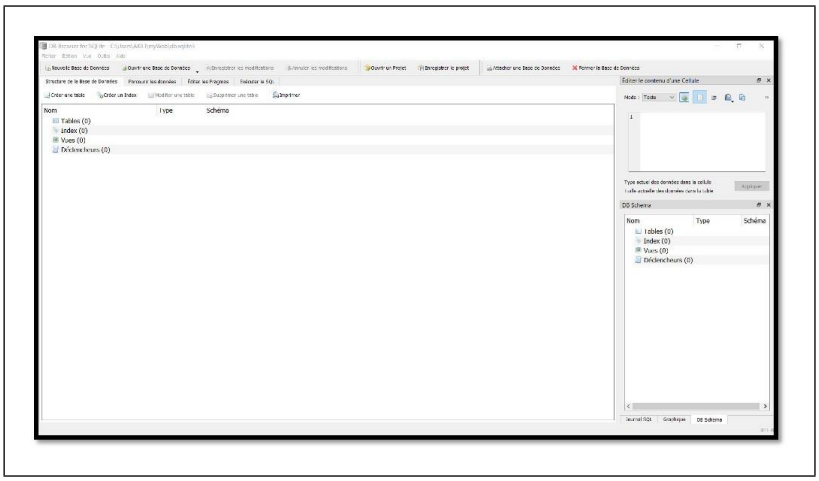


FIGURE 2.15 – Logiciel de lecture de base de données DB browser.

Pour importer les paramètres de fonctionnement de la plateforme à la base de données, on exécute la commande **python manage.py migrate** :

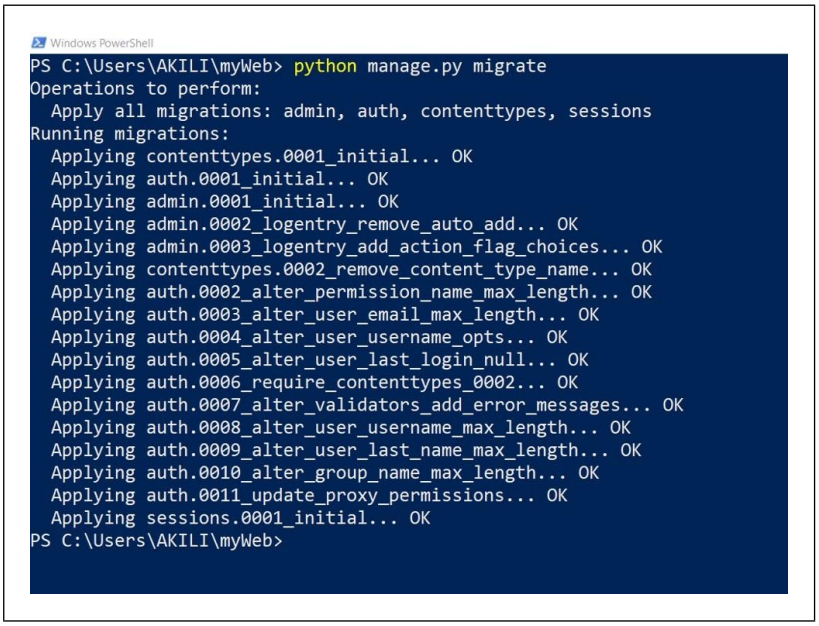


FIGURE 2.16 – La migration des tables de données crées vers le siteweb.

On ouvre la base de données et on trouve toutes les tables nécessaires pour la création d'un site.

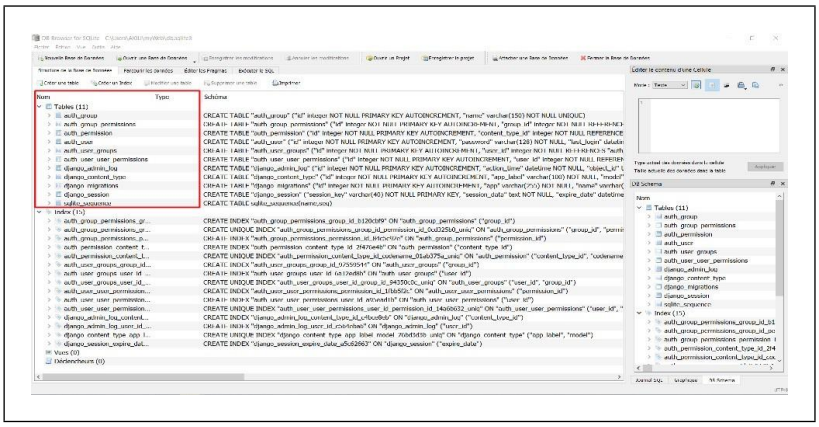


FIGURE 2.17 – Les tables de données crées par default du framework Django.

webPour créer un administrateur pour le projet Django on exécute la commande **python manage.py createsuperuser**, cette commande vas créer des champs pour créer un Nom d'utilisateur, un émail et un mot de passe :

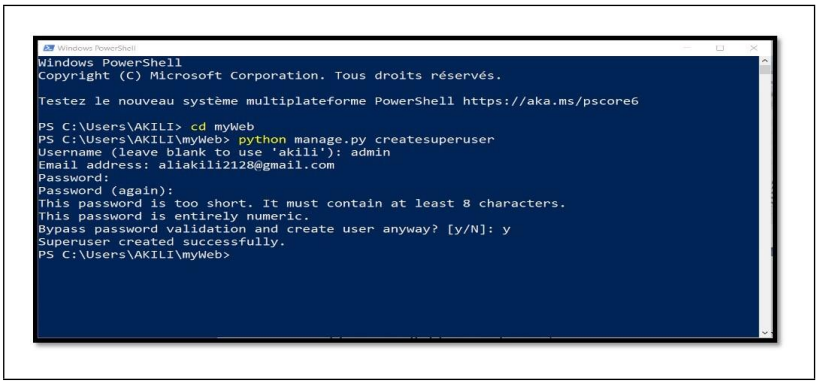


FIGURE 2.18 – Création d'un compte administrateur du projet Django.

On visualise la page d'administration du projet Django sur l'adresse 127.0.0.1/admin et ça donne :

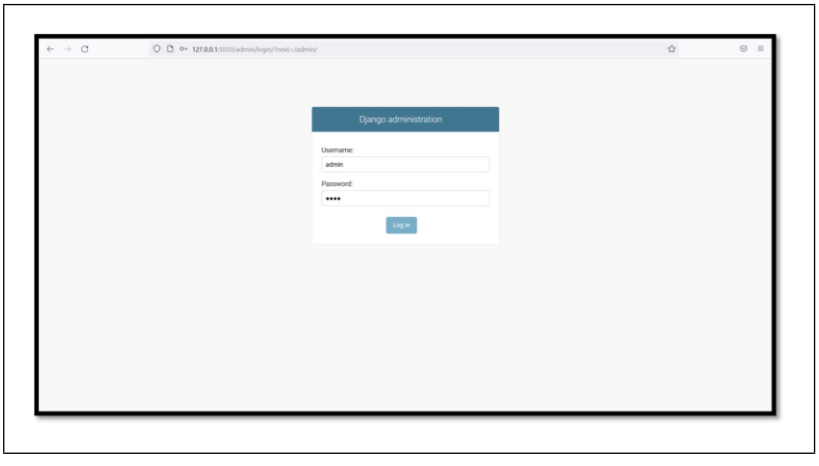


FIGURE 2.19 – Connexion du compte administrateur

En cliquant sur log in on accède directement vers page d'administration ou on peut créer et modifier les différentes données stockées sur la base de données.

FIGURE 2.20 – La table d'utilisateur dans la page d'administration du siteweb.

En suite on va sur la page Users et on trouves les coordonnés de l'admin créé.

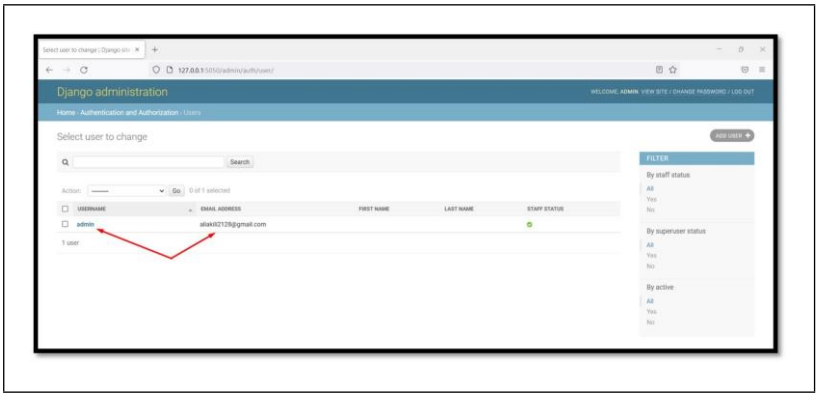


FIGURE 2.21 – Les cordonnées d'adlinistrateur créé.

2.7.4 La réalisation de l'application (le site web)

Dans cette partie on utilise PyCharm qui est un environnement de développement intégré à base de python pour gérer le projet Django et créer notre application web :

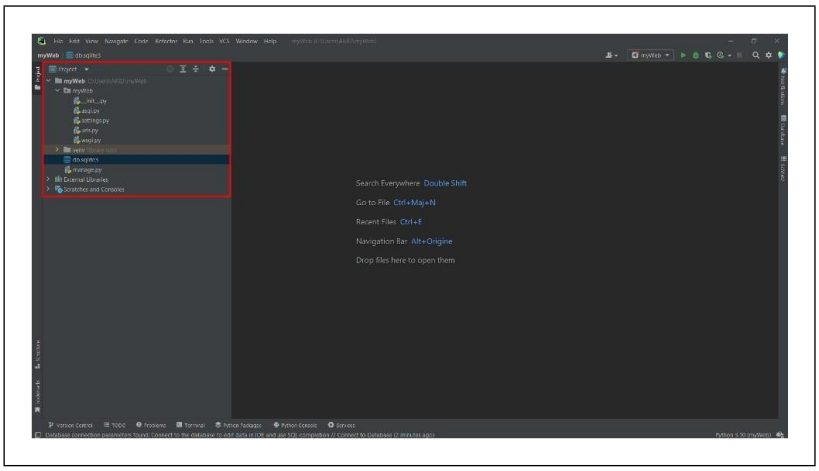


FIGURE 2.22 – L'exécution du projet Django avec PyCharm.

On tape la commande **python manage.py startapp myapp** et on remarque que le dossier de l'application est créé dans le dossier du notre projet :

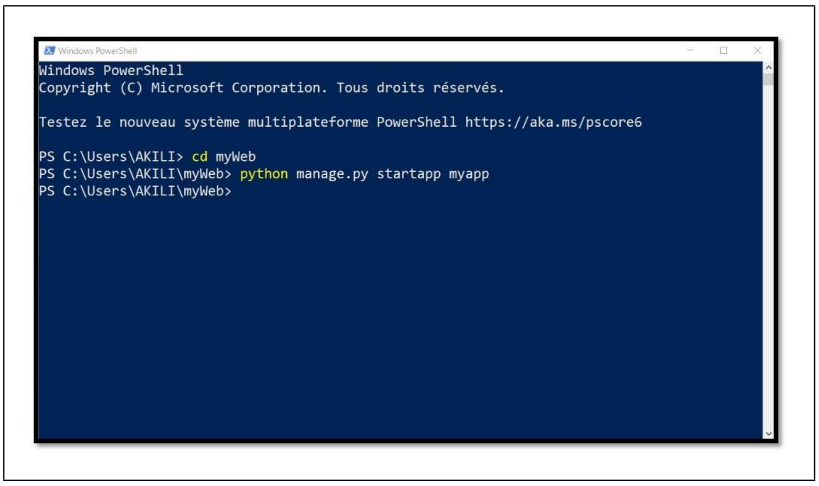


FIGURE 2.23 – La création de l'application.

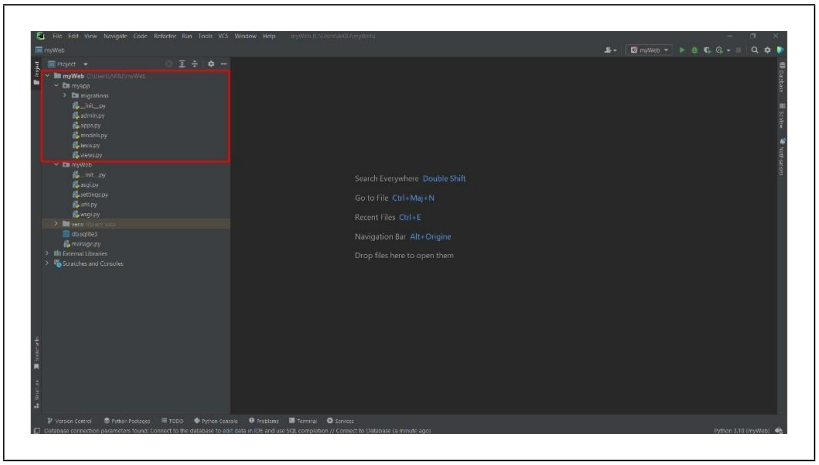


FIGURE 2.24 – Les fichiers Paython crées avec l'application.

Ensuite on intègre un template html du site qu'on veut implémenter sur Django dans un dossier ''template" et tous les fichiers statiques (css,javascript, images) dans un dossier ''static" :

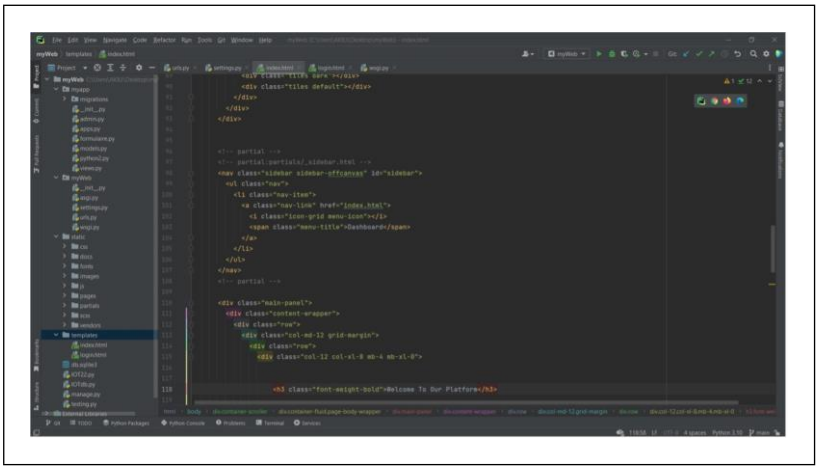


FIGURE 2.25 – L'importation du template HTML.

On fait la configuration pour chaque fichier python du l'application pour qu'elle puisse se connecter :

Models.py Models.py : dans ce fichier on crée les tables de données sous forme des class qui s'affiche sur la page d'administration du projet Django et la base de données :

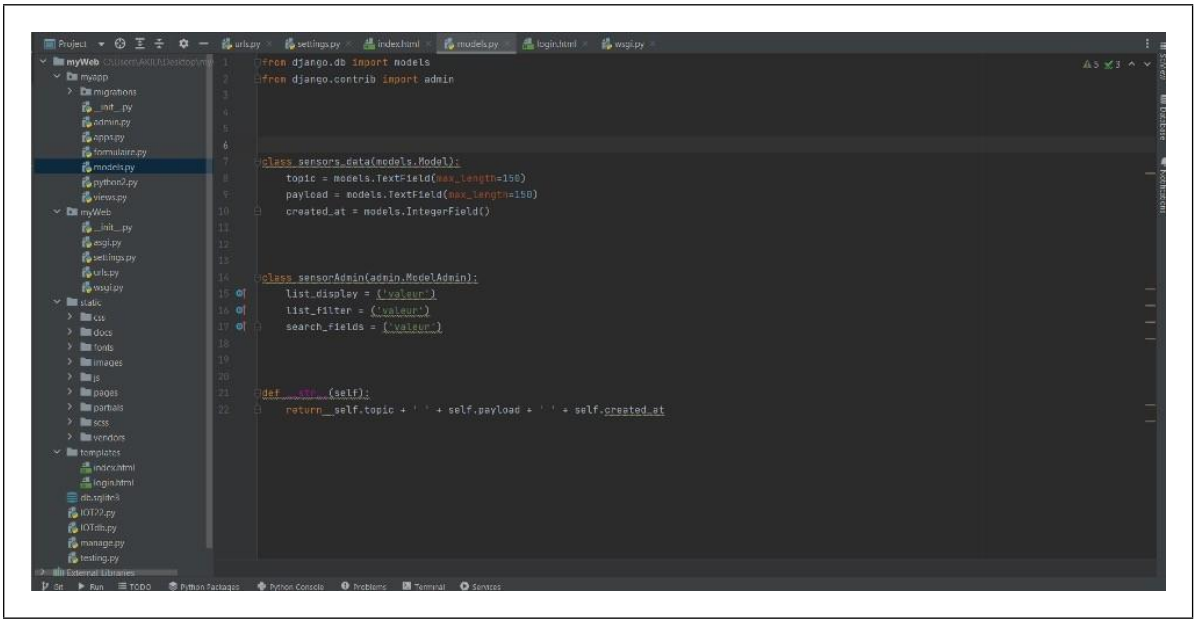


FIGURE 2.26 – La création de la table de données des capteurs.

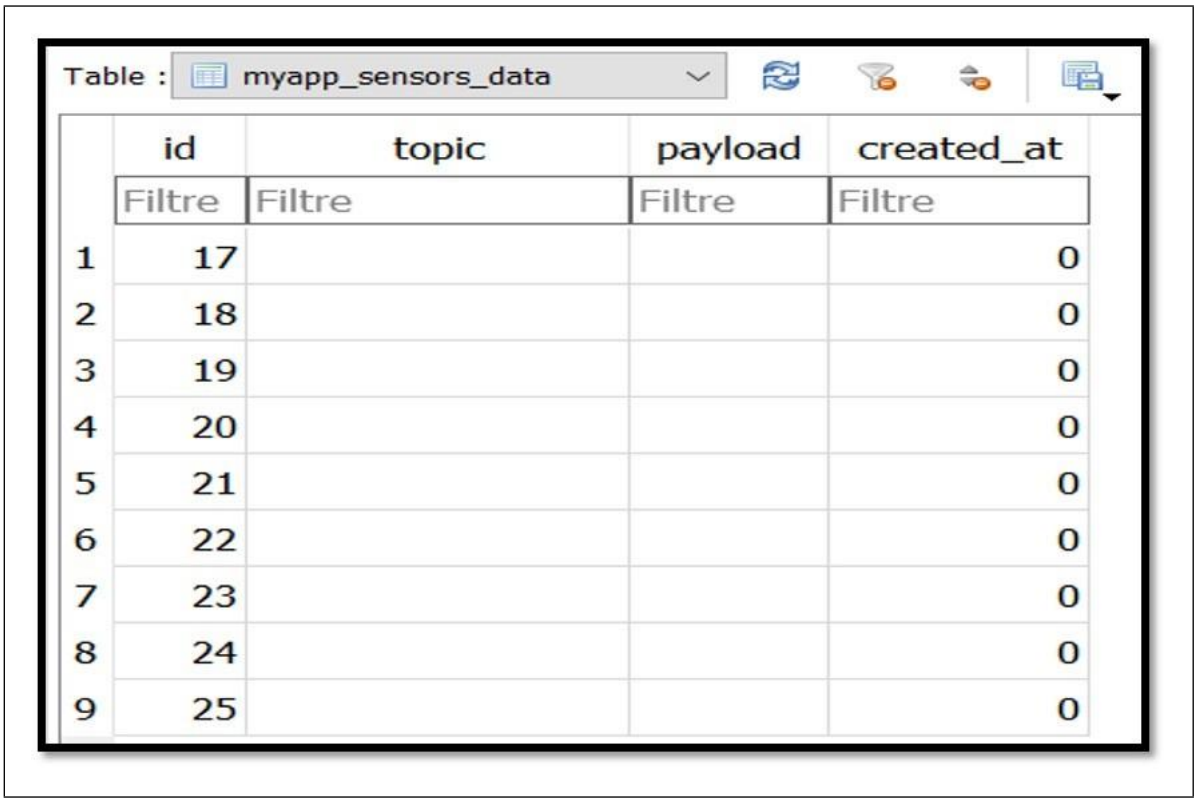


Table : myapp_sensors_data

	id	topic	payload	created_at
	Filtre	Filtre	Filtre	Filtre
1	17			0
2	18			0
3	19			0
4	20			0
5	21			0
6	22			0
7	23			0
8	24			0
9	25			0

FIGURE 2.27 – La table de données des capteurs.

Admin.py : ce fichier permet la page d'administration de lire et affichier les tables de la base de données, il autorise l'administrateur de gérer et modifier les contenus des tables au niveau du site Django en important les class de données du fichier **models.py**

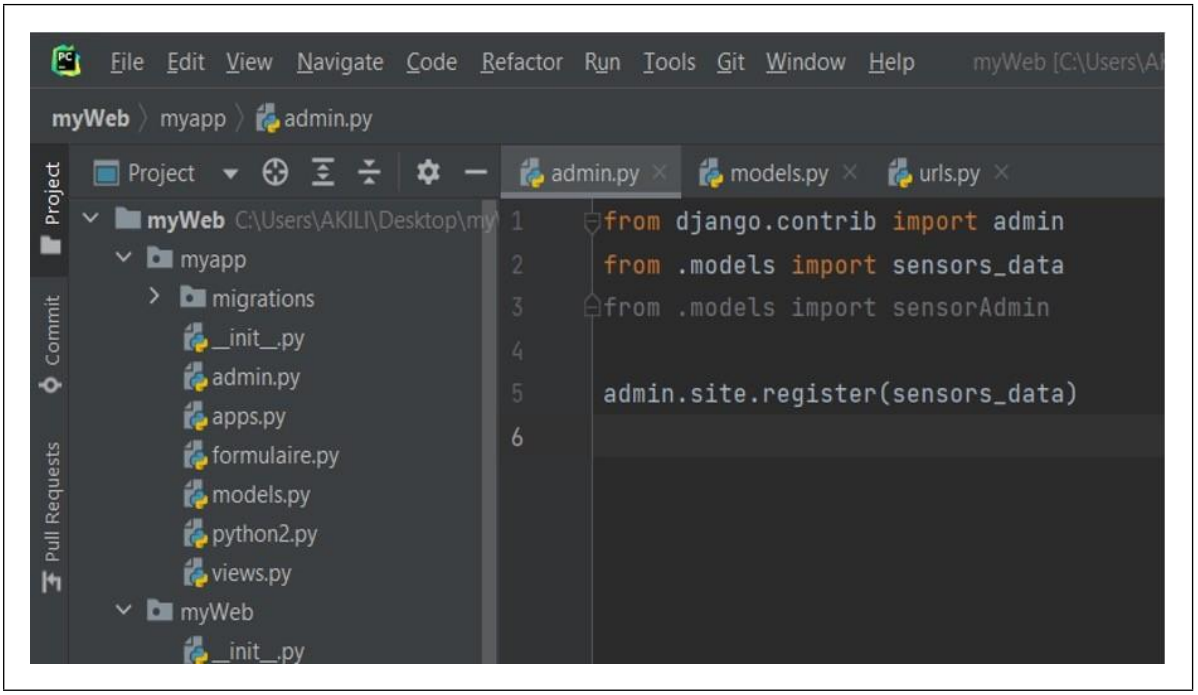


FIGURE 2.28 – L'importation de la table des capteur dans la page d'administration de Django.

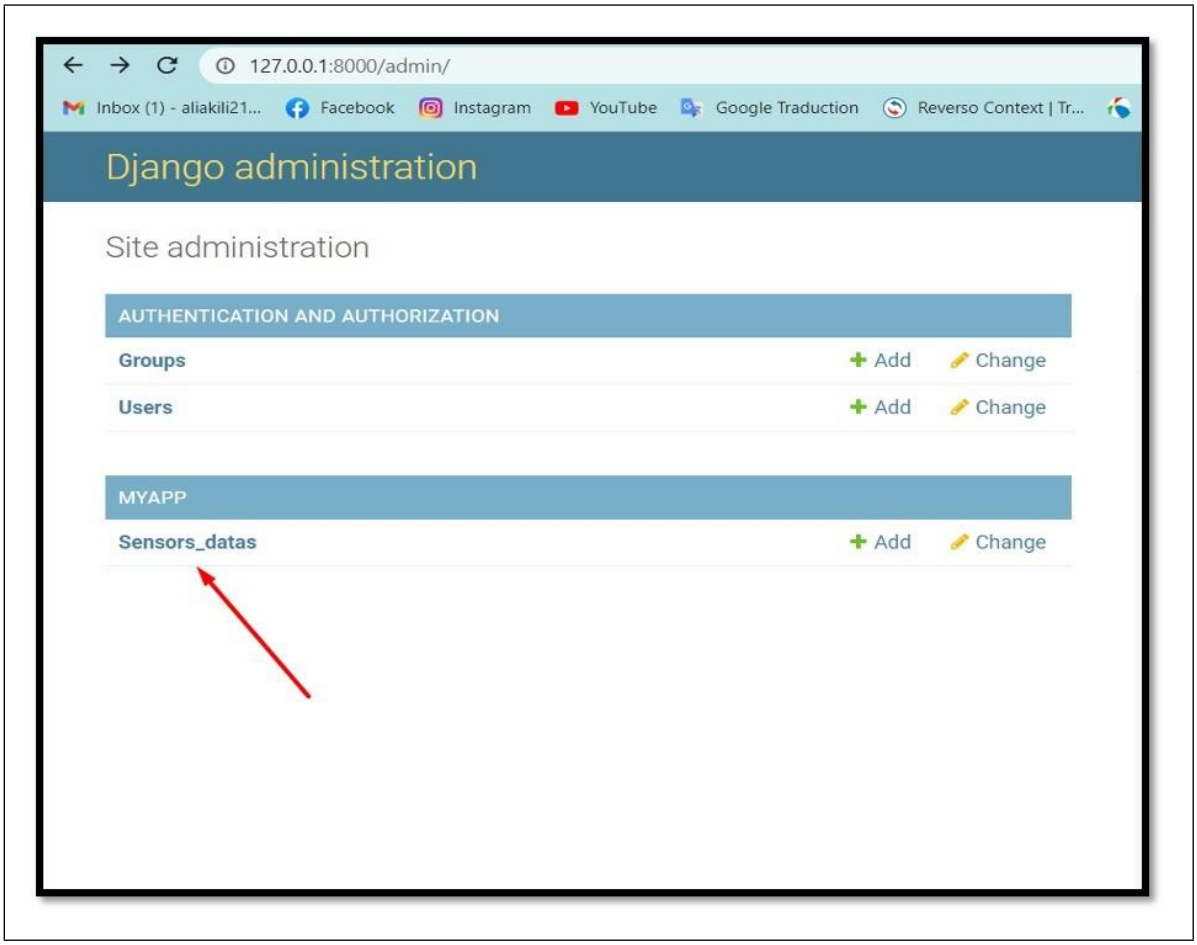


FIGURE 2.29 – La table de capteur dans la page d'administration Django.

Views.py : c'est le fichier responsable de l'affichage des différentes pages web programmées sur Django, il contient les fonctions d'affichage des pages html.

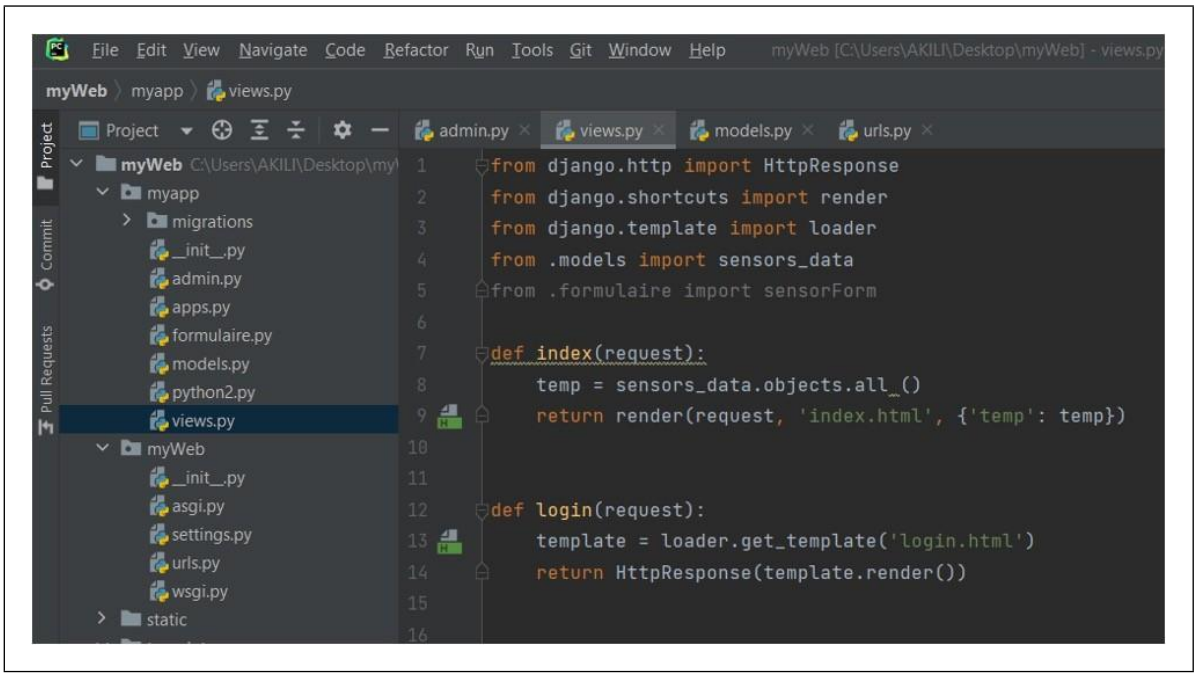


FIGURE 2.30 – Le fichier views.py

Urls.py : ce fichier génère des liens URLS pour chaque page qu'on veut afficher sur le site web :

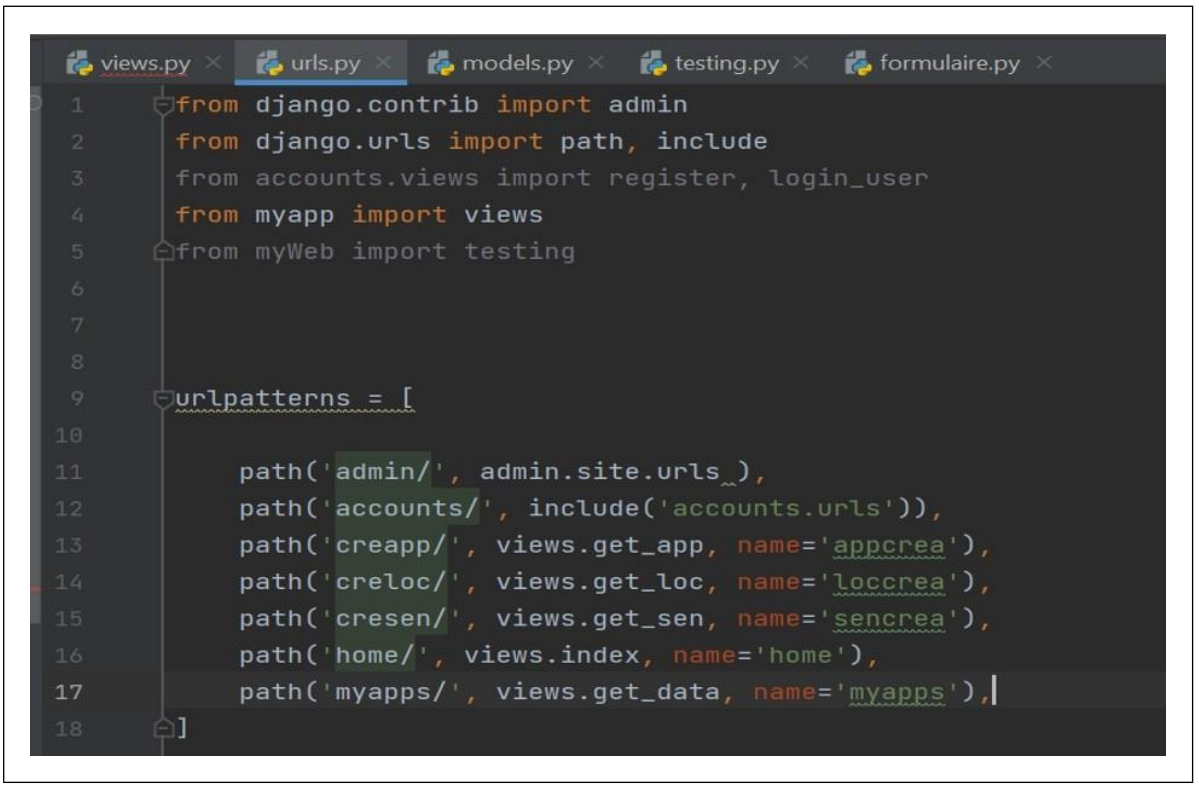


FIGURE 2.31 – Le fichier urls.py.

2.7.5 La création de l'application qui gère les comptes des utilisateurs

Cette application sert à créer des formulaires d'inscription et de connexion (register/login). Chaque personne qui veut utiliser les services de cette plateforme doit avoir un compte d'utilisateur avec un nom d'utilisateur et un mot de passe.

Le framework django contient des fonctions prédéfinies pour créer des formulaires ''register" et ''login" et les insérer dans le template html installé lors de la création de la plateforme

réation de l'application ''accounts '' :

Sur le terminal du PyCharm, on tape la commande suivante python manage.py startapp accounts, puis on crée les deux formulaires d'inscription et de connexion sur le fichier forms.py et les attribuer à des fonctions dans le fichier views.py Formulaire d'inscription et de connexion (register/login) :

forms.py

```python
from django import forms
from django.contrib.auth import get_user_model
from django.contrib.auth.models import User
from django.db import models
from django.contrib.auth.forms import UserCreationForm
non_allowed_usernames = ['abc']
# check for unique email & username
User = get_user_model()

class RegisterForm(forms.Form):
    username = forms.CharField()
    email= forms.EmailField()
    password1 = forms.CharField(
        label='password',
        widget=forms.PasswordInput(
            attrs={
                "class": "form-control",
                "id": "user-password"
            }
        )
    )
    password2 = forms.CharField(
        label='Confirm Password',
        widget=forms.PasswordInput(
            attrs={
                "class": "form-control",
                "id": "user-confirm-password"
            }
        )
    )
```

FIGURE 2.32 – Création du formulaire d'inscription d'utilisateur.

FIGURE 2.33 – Création du formulaire du connexion d'utilisateur.

L'exécution des formulaires d'inscription et de connexion se fait au niveau du fichier views.py dans la même application, ce dernier est relié à un fichier urls.py qui contient les liens pour accéder aux pages web.

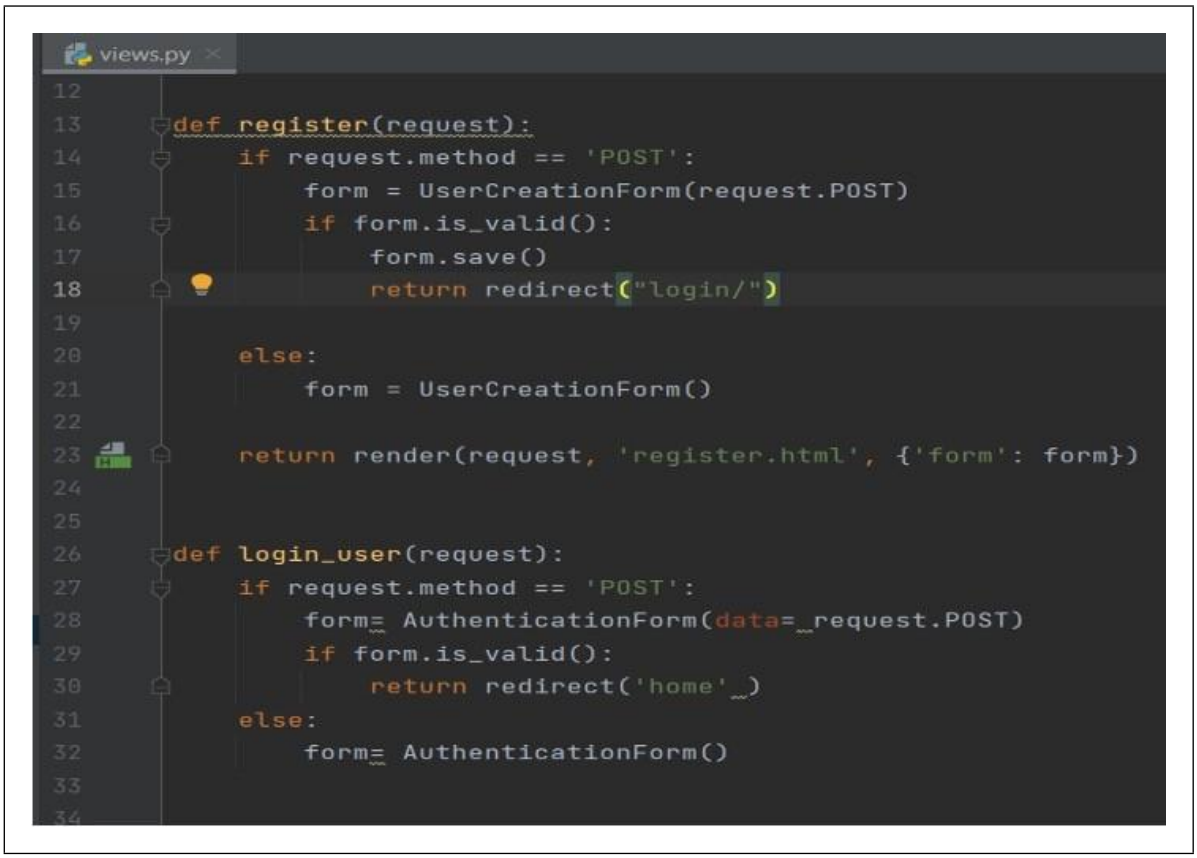


FIGURE 2.34 – Les fonctions d'exécution des deux formulaires d'inscription et de connexion.

Sur le site web, l'affichage des deux formulaires se fait en tapant l'url : **http ://127.0.0.1 :8000/accounts/register/** pour la page d'enregistrement et l'url : **http ://127.0.0.1 :8000/accounts/login/** pour la page de connexion comme montré ci-dessus :

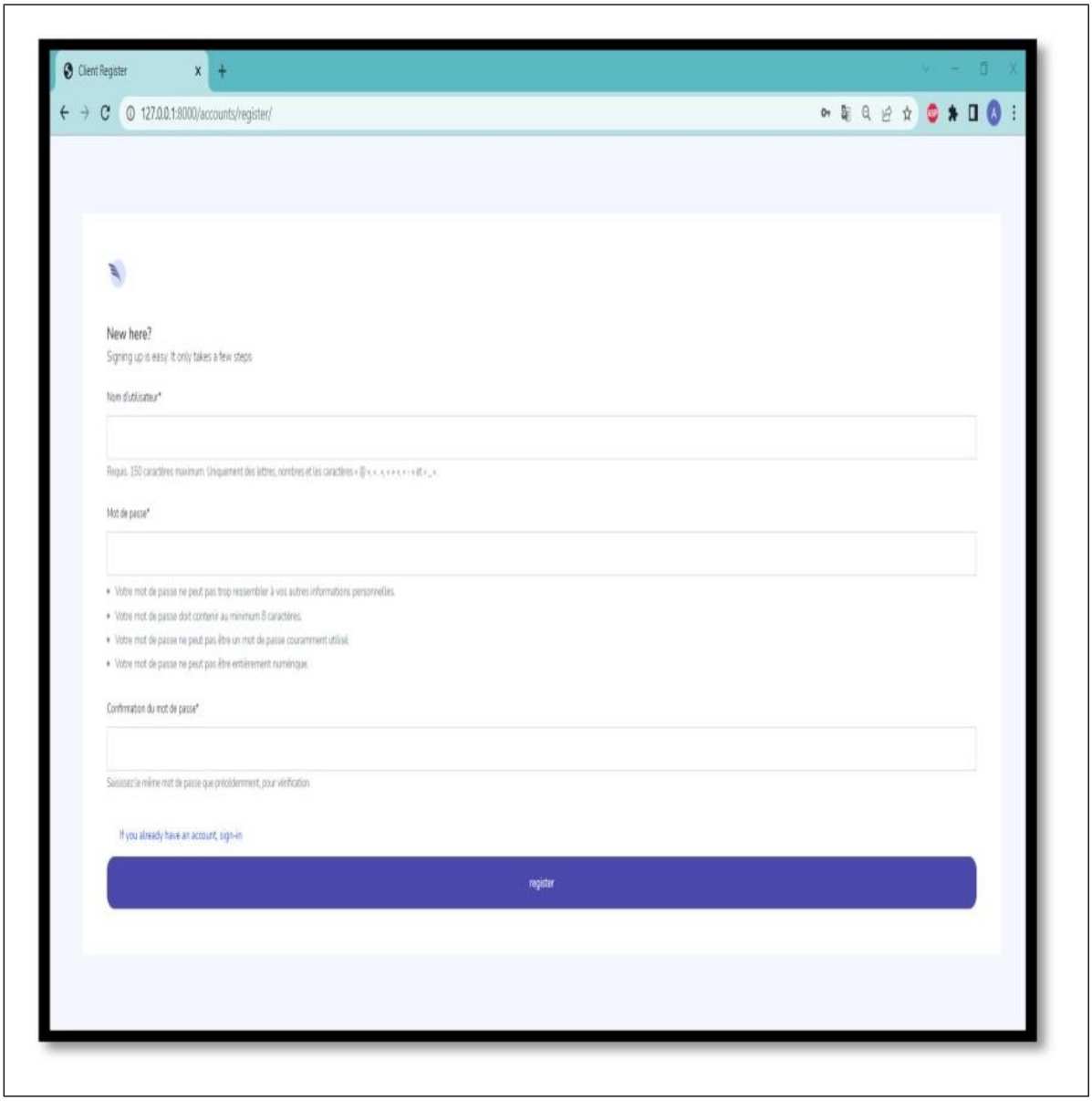


FIGURE 2.35 – L'affichage de la page d'inscription.

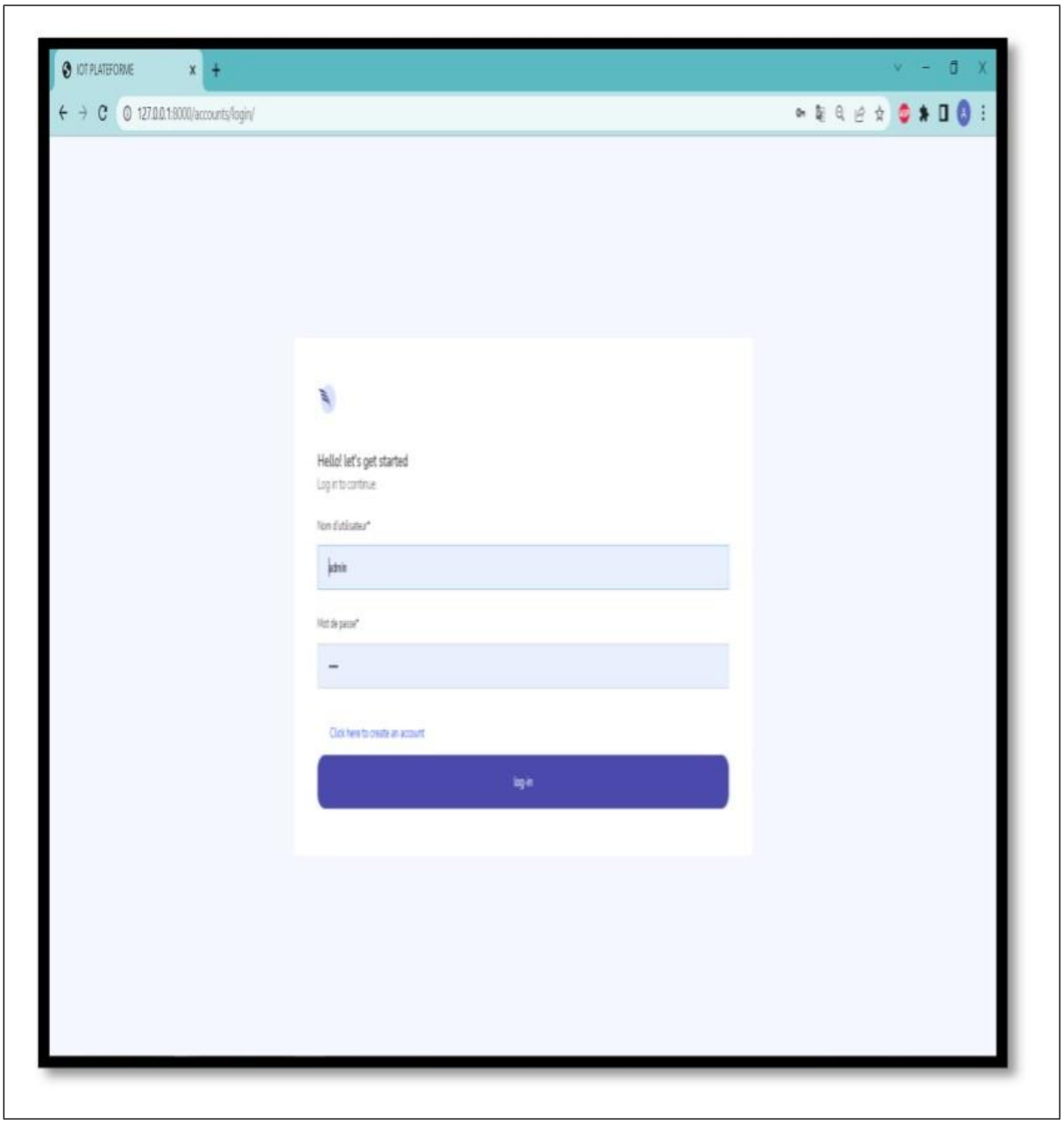


FIGURE 2.36 – L'affichage de la page de connexion.

2.8 La base de données

2.8.1 Définition

La création des tables de la base de données : On entend par base de données ou une Database une serie d'informations organisées pour faciliter leur accessibilité ; leur disponibilité ; leur gestion et leur mise à jour quotidienne par les utilisateurs.

Les differentes données d'une Database sont organisées et indexées en lignes, colonnes et tableaux pour faciliter la recherche. Les entreprises utilisent des bases de données pour le stockage, la gestion et la récupération des informations.

De façon générale, les bases de données sont exploitées pour la recherche des données collectée. Les utilisateurs saisient alors une requête. La base de données réside sur le

serveur avec une possibilité de la déplacer à n'importe quel moment. Les bases de données sont utilisées dans tous les domaines : les hôpitaux, les gouvernements, les compagnies aériennes, les assurances, l'éducation, Internet, la téléphonie, etc.

L'accès des utilisateurs aux bases des données doit être réglementer par ses administrateurs et ce afin de pouvoir pour contrôler le fonctionnement et l'utilisation. Les transactions de base de données doivent répondre aux exigences de conformité ACID.

2.8.2 La base de données SQLite

Maintenant, si l'utilisateur souhaite créer ou ajouter une application, il devra cliquer sur ≪Add a new Application≫ et il sera dirigé vers la page de création d'application : [34**?**]

2.8.3 La création des tables de la base de données

Afin que l'utilisateur puisse ajouter ses capteurs sur notre plateforme, Nous devons créer des tables dans la base de données pour stocker les données insérées par ce dernier.

La première étape consiste à créer une table pour les applications. Pour cela nous devonscréer notre modèle dans le fichier models.py pour informer la base de données du type de données qui seront insérées :

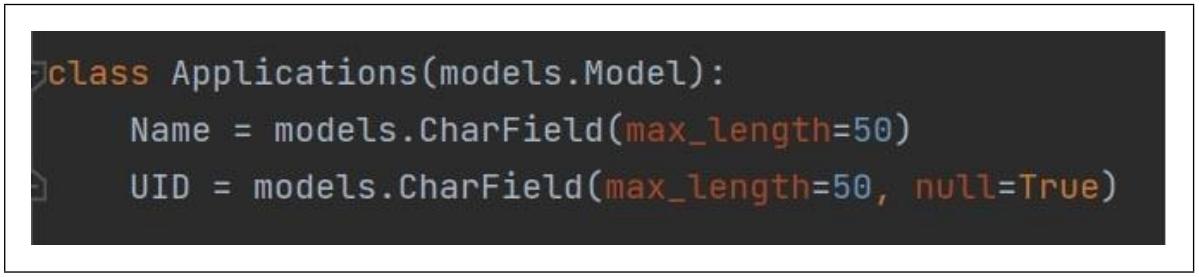


FIGURE 2.37 – Le modèle des Applications.

Pour contrôler cette classe depuis notre plateforme, nous avons besoin d'un formulaire que nous créons dans le fichier ≪formulaire.py≫ :

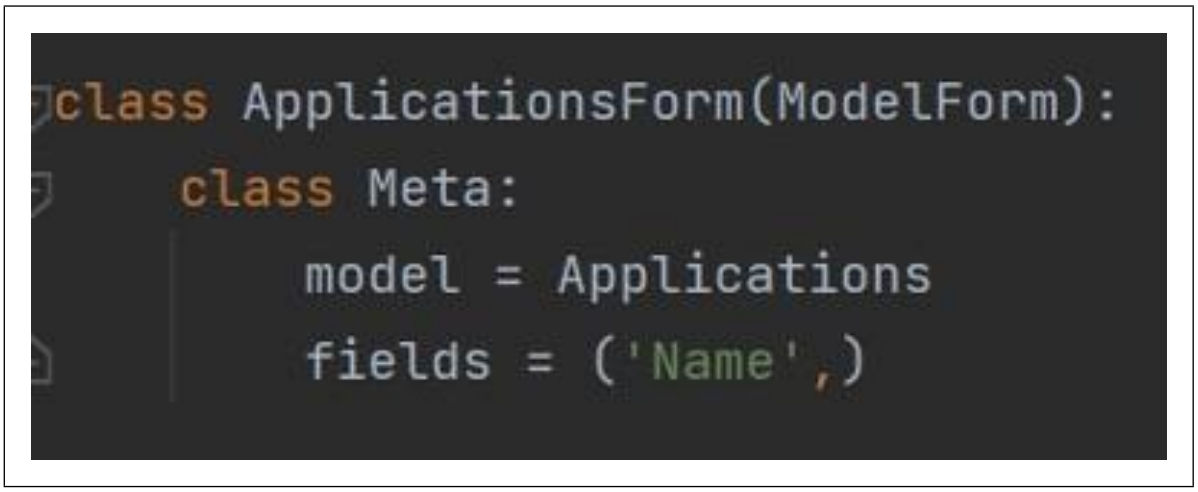


FIGURE 2.38 – Formulaire des Applications

La deuxième étape consiste à créer une table pour les locaux. Comme indiqué dans la première étape, nous aurons besoin d'un modèle et d'un formulaire :

FIGURE 2.39 – Le modèle des locaux

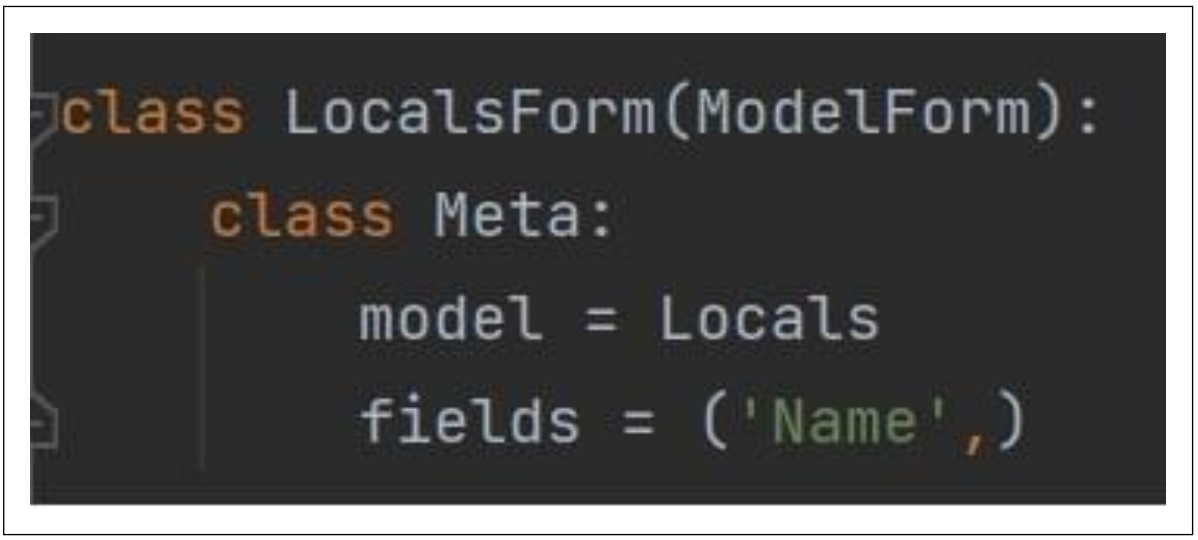


FIGURE 2.40 – Le formulaire des locaux

La troisième et dernière étape consiste à créer une table pour les capteurs. De manière similaire aux deux autres tables, nous aurons besoin d'un modèle et d'un formulaire :

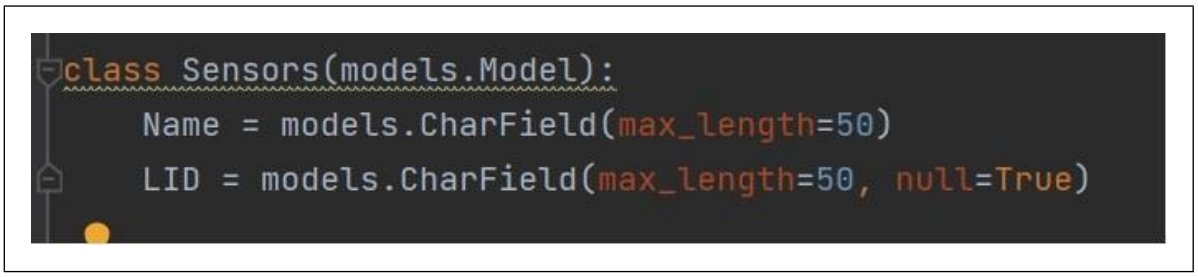


FIGURE 2.41 – Le Modèle des capteurs

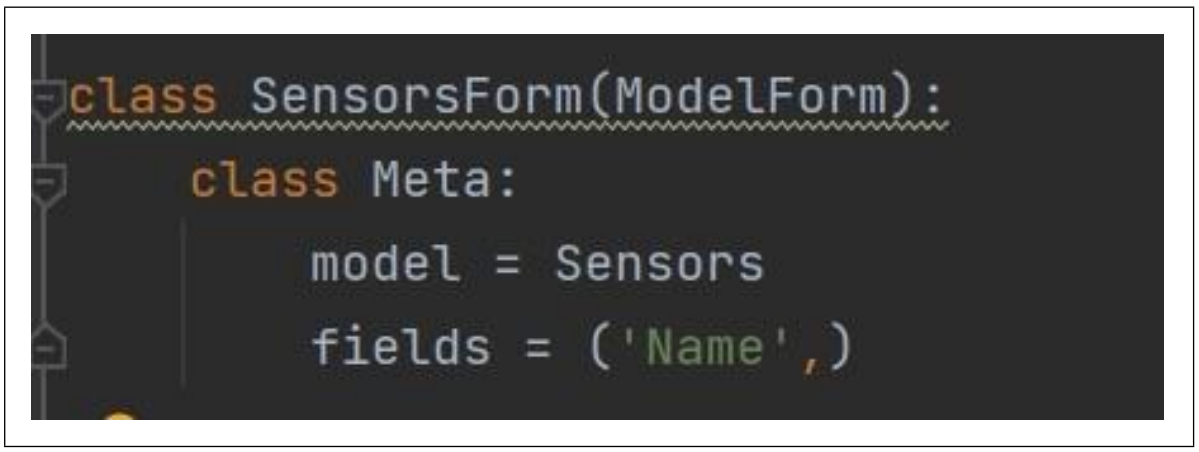


FIGURE 2.42 – Le formulaire des capteurs

Nous constatons à la fin que trois tables ont été créées dans notre base de données :

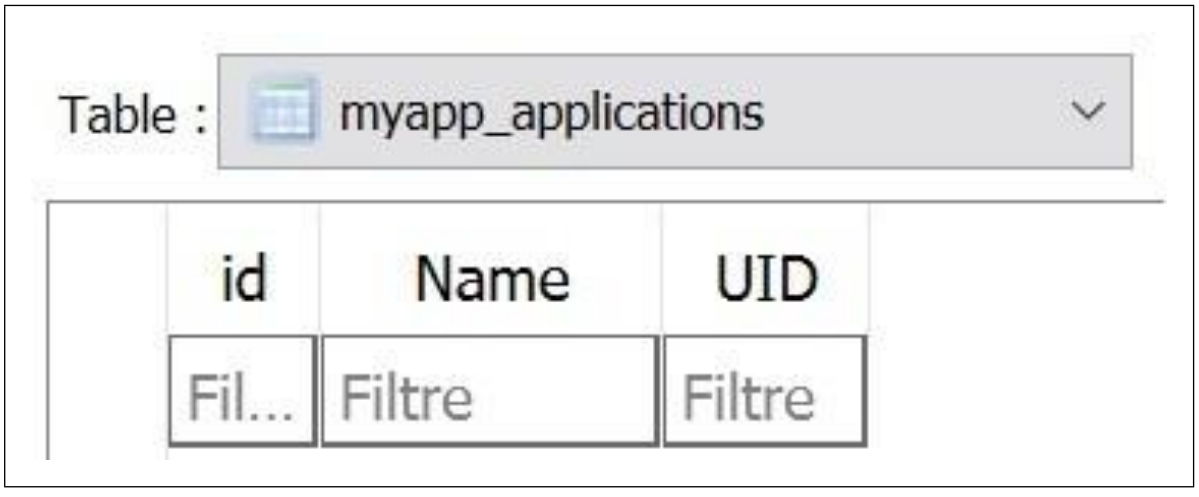


FIGURE 2.43 – Table des Applications

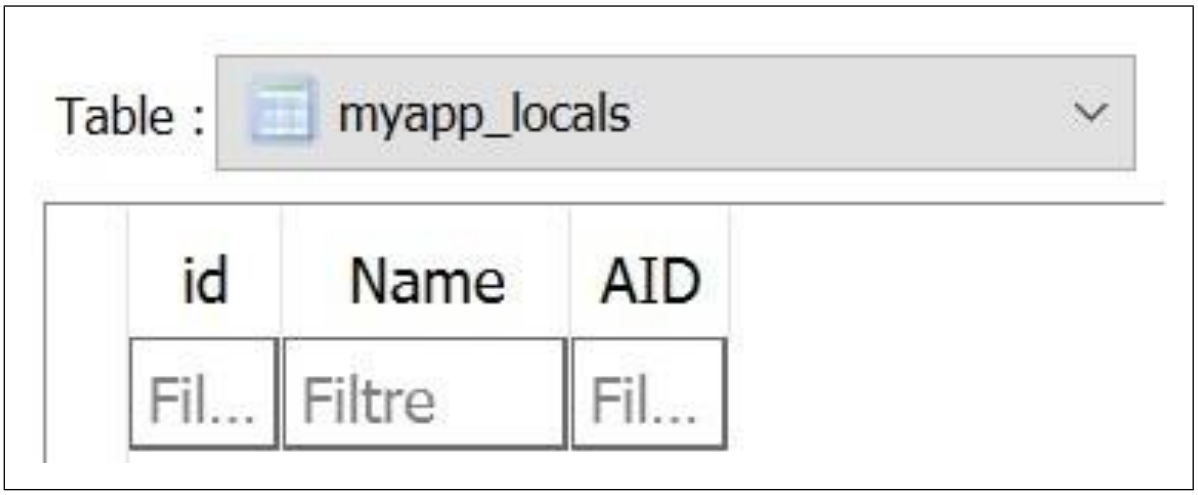


FIGURE 2.44 – Table des locaux

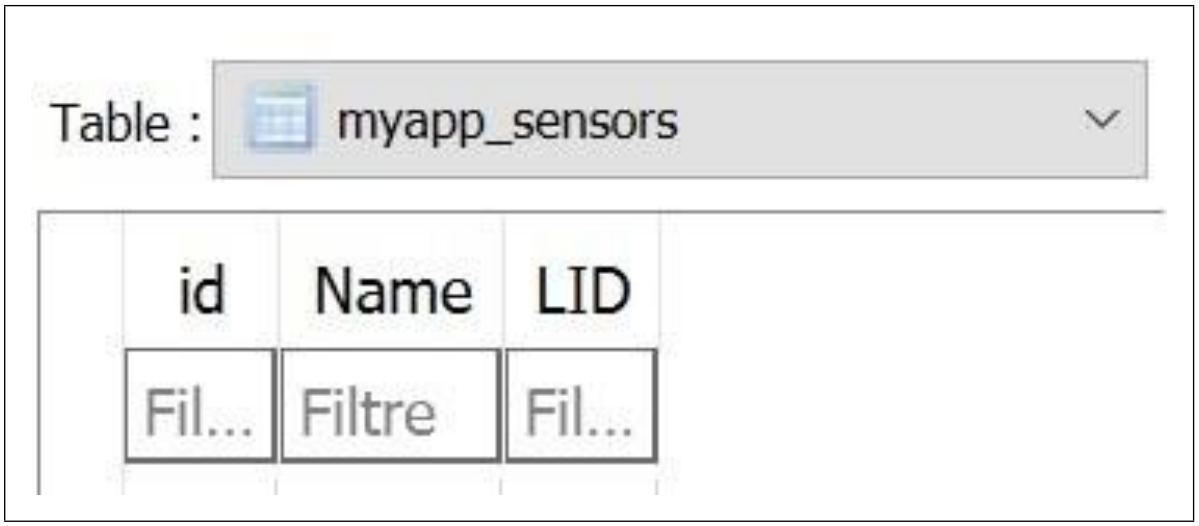


FIGURE 2.45 – Table des capteurs

2.8.4 Lien entre les tables

Pour bien organiser les données insérées par l'utilisateur, nous devons créer une relation ou un lien entre les tables de notre base de données.

La première étape consiste à relier la table des Applications avec notre utilisateur. Pour cela, nous avons écrit un script Python (dans le fichier views.py) qui va récupérer le «ID» de l'utilisateur qui crée l'application, et l'insérer dans le champ « UID » de la table des Applications. Nous pourrons ainsi savoir à quel utilisateur appartient l'application créée.

```python
def get_app(request):
    if request.method == 'POST':
        form = ApplicationsForm(request.POST)
        obj=form.save(commit=False)
        obj.UID=request.user.id
        obj.save()
        return redirect('loccrea')
    else:
        form = ApplicationsForm()
    return render(request, 'creapp.html', {'form': form})
```

FIGURE 2.46 – Lien entre l'application et l'utilisateur

La deuxième étape consiste à relier la table des locaux et la table des applications. Comme pour les applications, nous avons écrit un script qui va récupérer le « ID » de l'application à laquelle notre local va appartenir, et l'insérer dans le champ «AID» de la table des locaux. Nous pourrons ainsi savoir à quelle application appartient le local créé.

```python
def get_loc(request):
    if request.method == 'POST':
        form = LocalsForm(request.POST)
        loc = form.save(commit=False)
        ss=Applications.objects.last()
        loc.AID = ss.id
        loc.save()
        return redirect('sencrea')
    else:
        form = LocalsForm()
    return render(request, 'creapp2.html', {'form': form})
```

FIGURE 2.47 – Lien entre le local et l'application à laquelle il appartient

La troisième et dernière étape consiste à relier la table des capteurs à la table des locaux. De manière similaire aux deux premières étapes, nous avons écrit un script Python qui va récupérer le « ID » du local auquel appartient notre capteur, et l'insérer dans le

champ «LID» de la table des capteurs. Nous pourrons ainsi savoir à quel local appartient notre capteur.

```
def get_sen(request):
    if request.method == 'POST':
        form = SensorsForm(request.POST)
        sen = form.save(commit=False)
        sn = Locals.objects.last()
        sen.LID = sn.id
        sen.save()
        return redirect('home')
    else:
        form = SensorsForm()
    return render(request, 'creapp3.html', {'form': form})
```

FIGURE 2.48 – Lien entre le capteur et le local auquel il appartient

2.9 MQTT (Message Queuing Telemetry Transport)

Il s'agit d'un protocole de messagerie léger, à utiliser dans les cas où les clients ont besoin d'une faible empreinte de code et sont connectés à des réseaux non fiables ou à des réseaux dont les ressources en bande passante sont limitées. Il est principalement utilisé pour les communications de machine à machine (M2M) ou les connexions de type "Internet des objets".

2.9.1 Principe de fonctionnement

MQTT est basé sur le protocole TCP/IP en utilisant une topologie PUBLISH/SUBSCRIBE. Dans l'architecture MQTT, il existe deux types de systèmes : les clients et les courtiers :

Courtier (Broker) : C'est le serveur avec lequel les clients communiquent. Il reçoit les messages (ou data) des clients et les envoie aux autres clients.

Clients : On a deux cas de clients, le publieur, qui est chargé d'envoyer les messages au courtier, et on a l'abonné (ou subscriber) qui récupère les messages du courtier en s'abonnant à un sujet (topic).

Sujet (Topic) : L'endroit où les clients peuvent placer leurs messages, ou bien récupérer les messages d'autres clients.

MQTT est un protocole piloté par les événements. Il n'y a pas de transmission de données périodique ou continue. La transmission est ainsi réduite au minimum. Un client ne publie que lorsqu'il y a des informations à envoyer, et un courtier n'envoie des informations aux abonnés que lorsque de nouvelles données arrivent.

Une autre façon pour MQTT de minimiser ses transmissions est de construire des messages de petite taille et étroitement définis. Chaque message a un en-tête fixe de seulement 2 octets. Trois niveaux de qualité de service (QoS) différents permettent aux concepteurs de réseaux de choisir entre la minimisation de la transmission de données et la maximisation de la fiabilité.

— QoS 0 - Offre la quantité minimale de transmission de données. Avec ce niveau, chaque message est livré à un abonné une seule fois, sans confirmation. Il n'y a donc aucun moyen de savoir si les abonnés ont reçu le message. Comme ce niveau suppose que la livraison est complète, les messages ne sont pas stockés pour être livrés à des clients déconnectés qui se reconnectent par la suite.
— QoS 1 - Le courtier tente de remettre le message et attend une réponse de confirmation de l'abonné. Si une confirmation n'est pas reçue dans un délai donné, le message est envoyé à nouveau. Avec cette méthode, l'abonné peut recevoir le message plus d'une fois si le courtier ne reçoit pas la confirmation de l'abonné à temps.
— QoS 2 - Le client et le courtier utilisent une poignée de main en quatre étapes pour s'assurer que le message est reçu, et qu'il n'est reçu qu'une seule fois.

Pour les situations où les communications sont fiables mais limitées, la QoS 0 peut être la meilleure option. Pour les situations où les communications ne sont pas fiables, mais où les connexions ne sont pas aussi limitées en ressources, la QoS 2 serait la meilleure option. La QoS 1 fournit une sorte de solution optimale pour les deux mondes, mais exige que l'application recevant les données sache comment traiter les doublons.

Pour la QoS 1 et la QoS 2, les messages sont enregistrés ou mis en file d'attente pour les clients qui sont hors ligne et qui ont une session persistante établie. Ces messages sont renvoyés (selon le niveau de QoS approprié) lorsque le client est de nouveau en ligne.

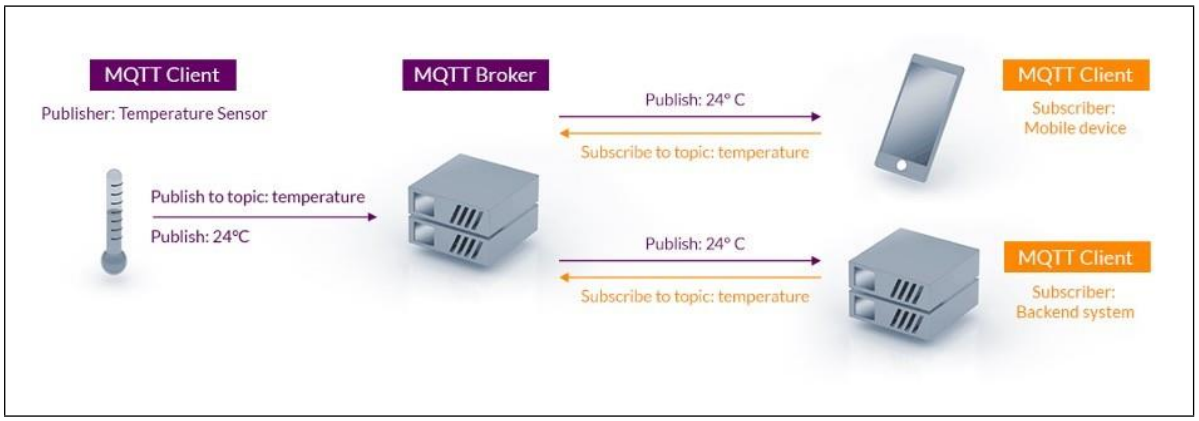


FIGURE 2.49 – Schéma explicatif du fonctionnement de MQTT. (www.mqtt.org)

2.9.2 Avantages du MQTT

Le protocole MQTT propose de nombreux avantages pour les connexions de type IOT, on cite à titre d'exemple les points suivants :

— Légèreté et efficacité : Les clients MQTT sont très petits, ils nécessitent donc des ressources minimales. Les en-têtes des messages MQTT sont petits pour optimiser la bande passante du réseau.

— Communications bidirectionnelles : MQTT permet d'envoyer des messages d'un dispositif au Cloud et du Cloud à un dispositif. Cela permet de diffuser facilement des messages à des groupes d'objets.
— Fiabilité : La fiabilité de la livraison des messages est importante pour de nombreux cas d'utilisation de l'IdO. C'est pourquoi MQTT a défini trois niveaux de qualité de service : 0 - au plus une fois, 1- au moins une fois, 2 - exactement une fois.
— Sécurité : MQTT permet de chiffrer facilement les messages à l'aide de TLS et d'authentifier les clients à l'aide de protocoles d'authentification modernes.

2.9.3 Mosquitto Broker

Mosquitto est un courtier (broker) open source et gratuit, qui travaille avec le protocole MQTT.

Mosquitto peut être installé sur n'importe quel système d'exploitation, et peut être utilisé depuis des microcontrôleurs comme l'ESP8266, mais il peut également être utilisé en ligne ou sur internet. Il suffit juste de saisir son adresse IP et le port que l'on souhaite utiliser dans le programme Arduino destiné à transférer les données collectées par les capteurs.

2.9.4 Eclipse Paho

Eclipse Paho est une implémentation open-source du client MQTT, disponible dans plusieurs langages de programmation, dans notre cas on a choisi le langage Python. Cette bibliothèque fournit une classe «client» qui permet aux applications de se connecter à un courtier MQTT (Mosquitto dans notre projet) pour publier des messages, s'abonner à des sujets et recevoir des messages publiés. Elle fournit également quelques fonctions d'aide pour rendre la publication de messages vers un serveur MQTT très simple.

Pour installer Eclipse Paho, il suffit juste de taper sur le terminal la commande suivante :

pip install paho-mqtt

On pourra par la suite utiliser la bibliothèque associée à Eclipse Paho ainsi que ses différentes classes et fonctions, dans notre programme Python.

Lien entre la base de données et le broker :

Afin de lier notre base de données avec le broker, on aura besoin de la bibliothèque Paho MQTT.

Dans un programme Python, on va importer cette bibliothèque ainsi que notre base de données, et saisir les coordonnées de notre broker.

Ensuite on devra créer deux fonctions qui permettent de se connecter au broker et de s'abonner au topic voulu, mais aussi d'attribuer les valeurs obtenues à l'aide des capteurs à une table dans notre base de données. Le programme ci-dessous fait cela :

```python
import paho.mqtt.client as mqtt
import sqlite3
from time import time

MQTT_HOST = 'test.mosquitto.org'       '''informations sur le broker'''
MQTT_PORT = 1883
MQTT_CLIENT_ID = 'Python MQTT client'

TOPIC = 'esp01/temperature'

DATABASE_FILE = 'db.sqlite3'


def on_connect(mqtt_client, user_data, flags, conn_result):    '''fonction pour se connecter au broker'''
    mqtt_client.subscribe(TOPIC)


def on_message(mqtt_client, user_data, message):
    payload = message.payload.decode('utf-8')  '''permet de récupérer les valeurs du topic'''

    db_conn = user_data['db_conn']
    sql = 'INSERT INTO myapp_sensors_data (topic, payload, created_at) VALUES (?, ?, ?)'
    cursor = db_conn.cursor() '''cursor pour la manipulation des données'''
    cursor.execute(sql, (message.topic, payload, int(time())))
    db_conn.commit()   '''pour confirmer que les changements ont été effectués'''
    cursor.close()

```

FIGURE 2.50 – Schéma explicatif du fonctionnement de MQTT. (www.mqtt.org)

Le programme suivant nous permettra de créer un client MQTT et d'exécuter les fonctions définies ci-dessus, afin de se connecter au broker, de s'abonner au topic, ainsi que récupérer les données et les placer dans la table de notre base de données

```python
def main():
    db_conn = sqlite3.connect(DATABASE_FILE)
    sql = """
    CREATE TABLE IF NOT EXISTS myapp_sensors_data (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        topic TEXT NOT NULL,
        payload TEXT NOT NULL,
        created_at INTEGER NOT NULL
    )
    """
    cursor = db_conn.cursor()
    cursor.execute(sql)
    cursor.close()

    mqtt_client = mqtt.Client(MQTT_CLIENT_ID)  '''création d'un client mqtt'''
    mqtt_client.user_data_set({'db_conn': db_conn})

    mqtt_client.on_connect = on_connect
    mqtt_client.on_message = on_message

    mqtt_client.connect(MQTT_HOST, MQTT_PORT)
    mqtt_client.loop_forever()


main()

```

FIGURE 2.51 – Schéma explicatif de la création d'un client mqtt

2.10 Conclusion

Dans ce chapitre, nous avons expliqué en détails les étapes de la réalisation de la plateforme, commençant par l'installation du Framework django et la création du la plateforme ainsi que le montage des capteurs avec la carte ESP. Nous avons aussi expliqué comment envoyer les données de capteurs vers le broker puis les stocker dans la base de donnée pour arriver à les afficher sur notre plateforme en utilisant le protocole MQTT.

Chapitre 3

Simulation de l’architecture et résultats

3.1 Introduction

Dans ce dernier chapitre, nous allons présenter les principales pages de notre plateforme et exposer les résultats des expérimentations de la connectivité des objets connectés avec la plateforme. Pour cela nous lançons notre serveur en exécutant la commande : **python manage.py runserver**, et nous nous connectons avec le compte que nous avons créé auparavant :

3.2 Les pages de la plateforme

3.2.1 Page de connexion

Pour se connecter à notre plateforme, l'utilisateur doit saisir son nom d'utilisateur et son mot de passe :

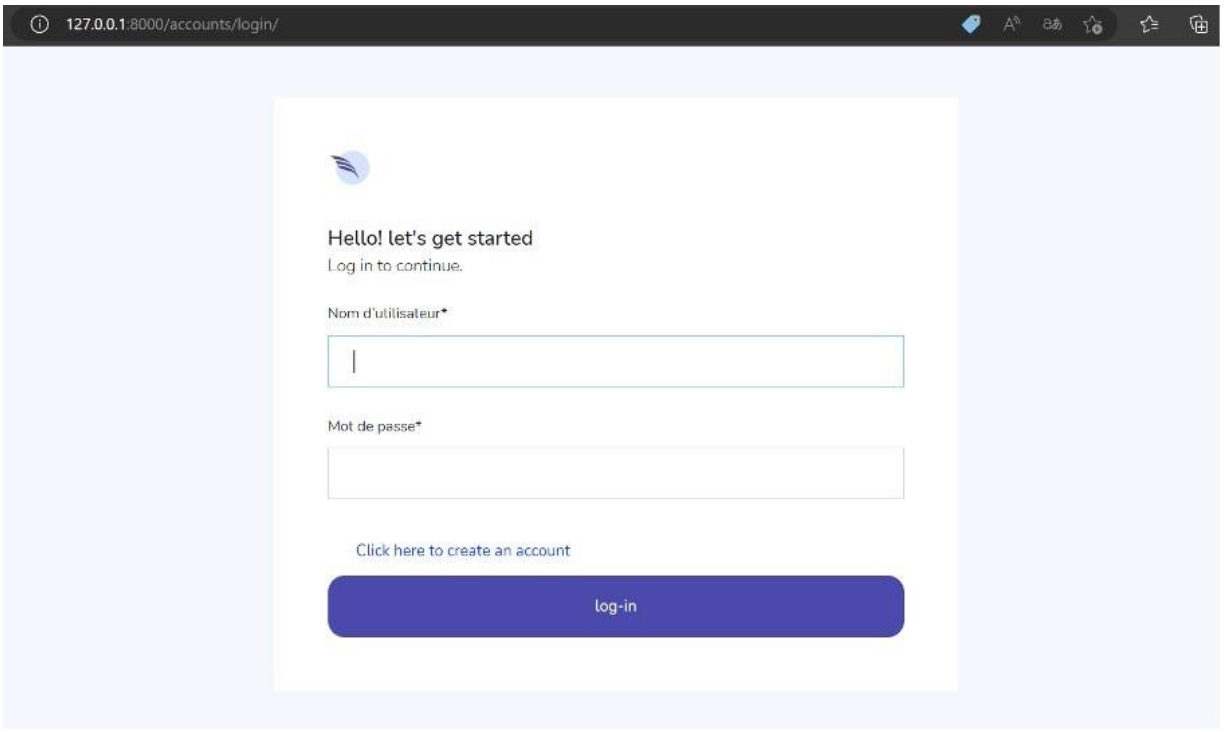


FIGURE 3.1 – Page de connexion.

3.2.2 Page d'inscription

Si l'utilisateur n'a pas encore de compte, il clique sur ≪Click here to create an account≫ et il sera dirigé vers la page d'inscription :

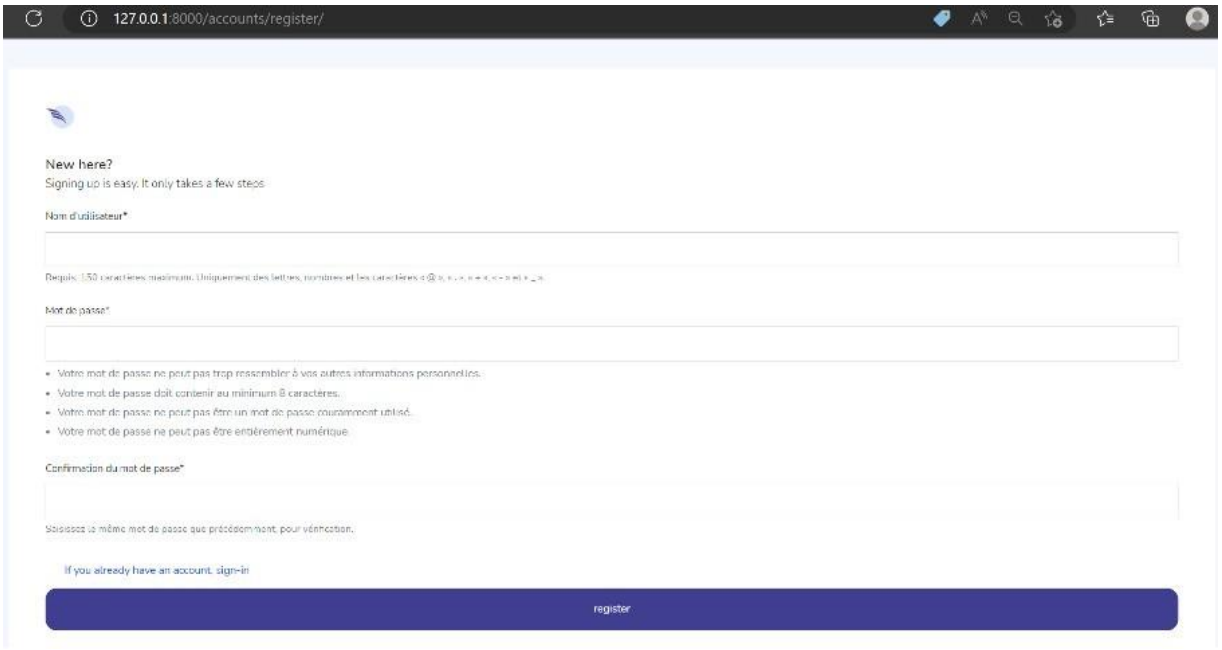


FIGURE 3.2 – Page d'inscription.

3.2.3 Page principale

L'utilisateur sera par la suite dirigé vers la page principale (home page) :

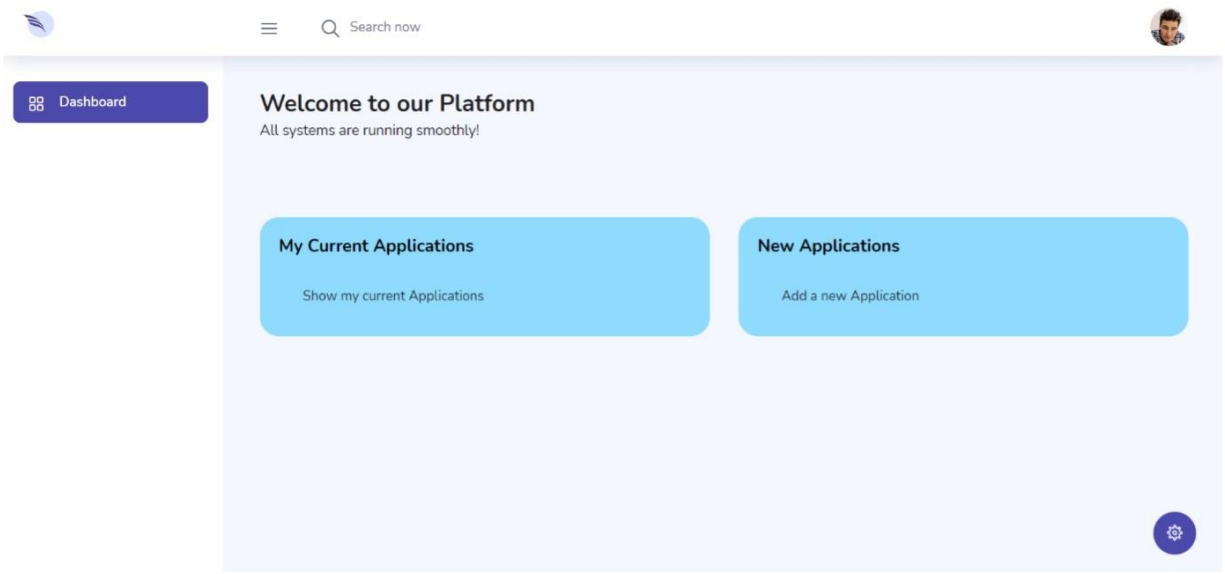


FIGURE 3.3 – Page principale.

3.2.4 Page de création d'application

Maintenant, si l'utilisateur souhaite créer ou ajouter une application, il devra cliquer sur «Add a new Application» et il sera dirigé vers la page de création d'application :

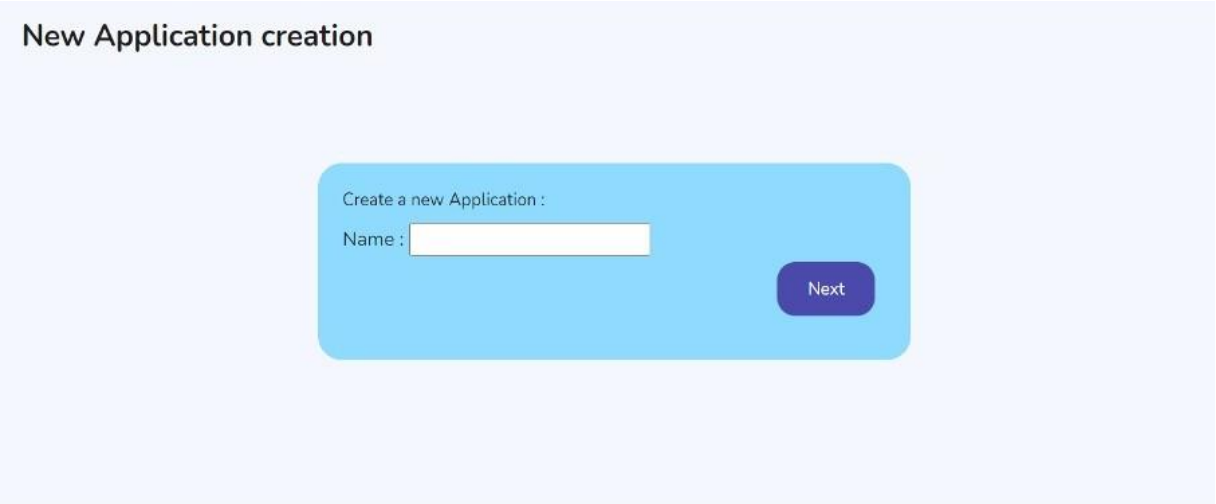


FIGURE 3.4 – Page de création d'une application.

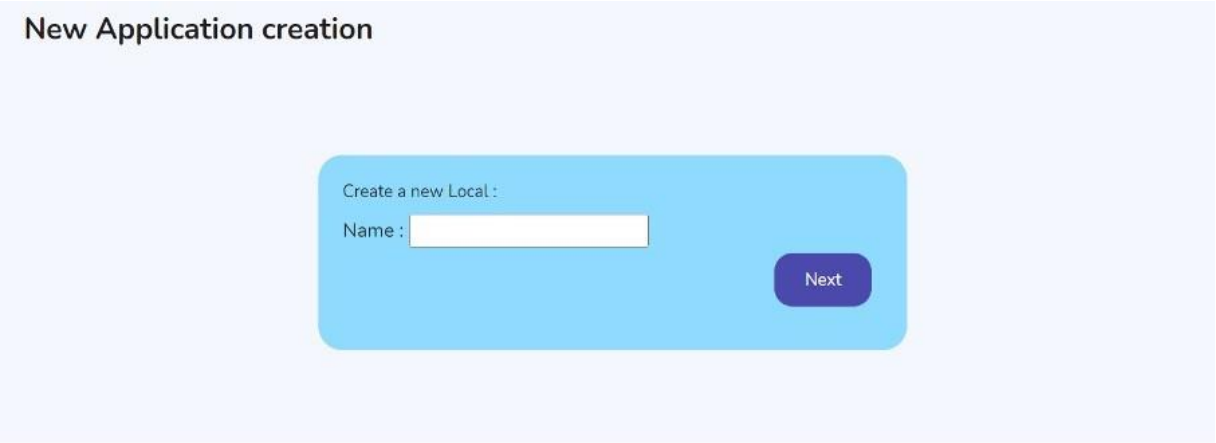


FIGURE 3.5 – Page de création d'un local

FIGURE 3.6 – Page de création d'un capteur

3.2.5 Page de visualisation des données

Si l'utilisateur souhaite visualiser les données des capteurs qu'il a créés, il doit cliquer sur « show my current applications », et il sera dirigé vers la page de visualisation des

données.

3.3 Exécution du programme Arduino

Dans cette partie nous avons utilisé l'IDE Arduino pour programmer la carte ESP8266 en langage C. Après le branchement du capteur avec l'ESP8266, et l'envoi du programme Arduino au microcontrôleur en utilisant un câble micro-usb, nous obtenons les résultats démontrés dans la figure ci-dessous :

FIGURE 3.7 – Envoi des données au Broker

Les connexions au réseau internet et au broker se sont donc bien déroulées.

Notre capteur affiche donc une température de 129,90°C et une humidité de 99,90%.

Nous utilisons à présent l'extension de Chrome MQTTLens, qui nous permet de nous connecter à notre broker et de nous abonner à nos topics, on constate les mêmes résultats :

FIGURE 3.8 – Visualisation des données sur MQTTLens

Maintenant, nous allons confirmer que les données ont bien été envoyées à la plateforme :

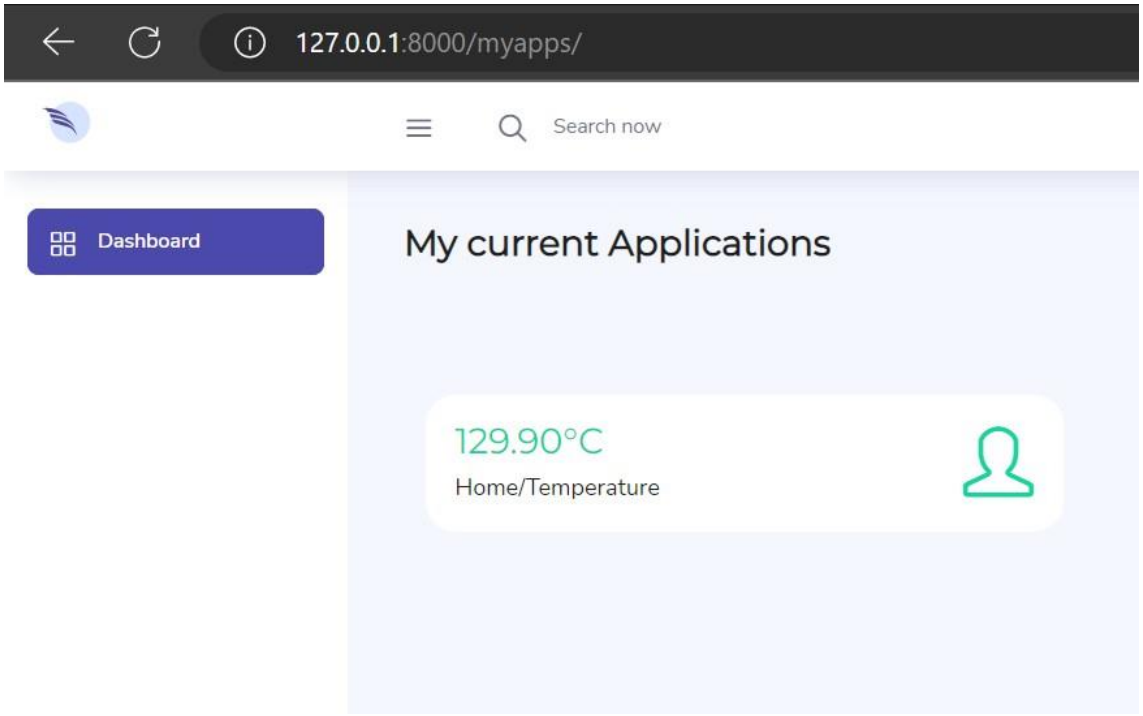


FIGURE 3.9 – Affichage de la donnée.

Effectivement, la donnée apparaît bel et bien dans notre plateforme, la connectivité entre cette dernière et le Broker fonctionne donc comme prévu.

3.4 Les problèmes rencontrés

Le principal problème que nous avons rencontré, est l'automatisation du script qui récupère les données du Broker. Ce dernier reste perfectible afin d'assurer une plus grande polyvalence à notre plateforme.

3.5 Conclusion

Dans ce chapitre, nous avons présenté les principales pages de notre plateforme, mais également exposé le fonctionnement de notre architecture IOT et testé la connectivité entre la carte ESP8266 et notre plateforme.

Conclusion générale

L'Internet des Objets est une industrie riche qui pourrait changer radicalement l'équation économique du marché en apportant de nouveaux services et en créant de nouvelles opportunités. Elle permet également de répondre aux besoins des particuliers, et ce par le développement de nouvelles applications permettant la réalisation de certaines taches. Dans notre projet, il a été question de développer une application web pour gérer des objets connectés distants depuis l'interface graphique de notre plateforme.

Nous avons donc créé une plateforme de gestion d'un réseau de capteurs en utilisant le langage de programmation Python, ainsi que son framework Django, qui permet de développer des applications web fluides et interactives.

Cette plateforme a été créée dans le but de contrôler et collecter des données issues de capteurs distants, en utilisant un serveur broker en ligne comme intermédiaire entre la partie logicielle et la partie matérielle. Le transfert de données s'effectue grâce au protocole MQTT qui est un protocole de messagerie de type publication/souscription basé sur le protocole TCP/IP.

Dans la partie matérielle, nous avons réalisé un réseau de capteurs reliés par une carte ESP8266, qui est connectée à notre réseau mobile afin d'envoyer les données issues des capteurs à notre serveur Broker.

Lors de la réalisation de ce projet, nous avons eu l'opportunité d'acquérir de nouvelles connaissances et de nouvelles compétences. Nous avons appris à travailler avec le langage de programmation Python ainsi que son Framework Django, le protocole MQTT, mais également le langage SQL pour les bases de données. Nous avons notamment eu l'opportunité d'appliquer certaines de nos connaissances acquises durant notre parcours universitaire, comme les langages de développement web dans la partie front-end de notre plateforme, et le langage Arduino dans la partie matérielle de notre architecture.

Nous avons pu atteindre le but de ce projet, qui est de réaliser une plateforme de collecte et de stockage de données issues de capteurs distants.

Cependant, ce projet reste encore perfectible pour rendre notre plateforme plus polyvalente. Pour cela nous avons besoin d'améliorer ou de développer notre script Python, responsable de la collecte des données stockées dans le serveur Broker, afin de rendre le tri de ces dernières et leur insertion dans les tables de la base de données plus dynamique.

Bibliographie

[1] Zineddine Kouahla. Plateforme de développement pour l'internet des objets (ido) avec un apprentissage automatique. 2019.

[2] Pierre Lemonnier. Et pourtant ça vole ! l'ethnologie des techniques et les objets industriels. *Ethnologie française*, pages 17–31, 1996.

[3] Michel Dahan. Une guerre économique d'une violence inédite. ***L****e journal de l'ecole de Paris du management*, 107(3) :36–42, 2014.

[4] Pierrick Larose et al. L'influence du concept d'industrie 4.0 sur les activités du champ industriel wallon. 2022.

[5] Sandrine Caroly. Activité collective et réélaboration des règles comme ressources pour la santé psychique : le cas de la police nationale. ***L****e travail humain*, 74(4) :365–389, 2011.

[6] Oussama Meski. *Développement d'un outil à base de connaissances pour l'aide à la décision dans le contexte de l'Industrie 4.0 : application au diagnostic des machines d'usinage à grande vitesse*. PhD thesis, Nantes, 2021.

[7] Nasreddine Bouhaï and Imad Saleh. *Internet des objets : Évolutions et Innovations*. ISTE Group, 2017.

[8] Soulef Benmakhlouf, Manel Amarouche, and Yamina Chiha. Commande intelligente de l'éclairage d'une maison. 2020.

[9] Abderrahmen Rafai and Sofia Kouah. Développement d'un système d'iot (internet of things) pour le smart lighting sous la plateforme ibm. 2018.

[10] Adel ROZTANE and Weam ALI MERINA. *Conception et réalisation d'un système embarqué pour la sécurité d'incendie.* PhD thesis, Directeur : Mr MEGNAFI Hicham/Co-Directeur : Mr BENNACER Djamel, 2022.

[11] Mohamed Ali Kandi. *Lightweight key management solutions for heterogeneous IoT*. PhD thesis, Compiègne, 2020.

[12] Sébastien Moret. Langues internationales, alphabets et révolution. les idées de nv jušmanov. *Studi slavistici*, 14 :275–292, 2017.

[13] Pierre Gillain and Didier Reynders. "la transformation digitale des opérateurs de télécommunication, quels développements et quels enjeux ? illustration avec orange belgium. 2019.

[14] Jean-Philippe Pernin. Objets pédagogiques : unités d'apprentissage, activités ou ressources. *Revue" Sciences et Techniques Educatives", Hors série*, pages 179–210, 2003.

[15] Alain Jeantet. Les objets intermédiaires dans la conception. eléments pour une sociologie des processus de conception. *Sociologie du travail*, pages 291–316, 1998.

[16] Pablo Iriarte and Isabelle De Kaenel. Les catalogues des bibliothèques : du web invisible au web social. 1ère partie : ouverture du catalogue à l'intégration des nouveaux contenus. *Revue Électronique Suisse de Science de l'Information*, (5), 2007.

[17] Pierre-Jean Benghozi, Sylvain Bureau, and Françoise Massit-Folea. L'internet des objets. quels enjeux pour les européens ? 2008.

[18] Bruno Bourassa and Fernand Serre. *Apprendre de son expérience*. Puq, 1999.

[19] Manel Brahmi and Hichem Laib. Générateur photovoltaïque connecté au réseau électrique et doté d'un filtre actif parallèle (fap). 2020.

[20] Nahit Pawar. *Conception d'une architecture complète pour l'interopérabilité des objets connectés hétérogènes et des services de l'Internet des Objets*. PhD thesis, Institut Polytechnique de Paris, 2021.

[21] Samia Tebib and Soualah Abdelghani. *La Sécurité Des Données Dans L'Internet Des Objets (IoT)*. PhD thesis, Universite laarbi tebessi tebessa, 2017.

[22] David Fettig. *Transferts des couples chaleur-masse dans les systèmes constructifs bois : traitement et exploitation de données expérimentales*. PhD thesis, Université de lorraine, 2017.

[23] Athanase Mawugnon ATCHOME, Eugène C EZIN, Théodore MY TAPSOBA, Claude LISHOU, Marc K ASSOGBA, Michel DOSSOU, Antoine C VIANOU, Luigi Alfredo GRIECO, and Thierry O EDOH. *Un modèle de réseau hybride intégrant les communications NB-IoT et D2D*. PhD thesis, EPAC/UAC, 2020.

[24] DAHMANE Abir BOUZIDI Meriem ELABED Sid et al. *Implémentation d'n Réseau Prototype pour l'Internet des Objets*. PhD thesis, Faculté des Sciences et Technologies, 2021.

[25] Fatma Merabet. *Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile*. PhD thesis, Université de Limoges ; Université Mouloud Mammeri (Tizi-Ouzou, Algérie), 2021.

[26] Hélène Sauzéon and Lucile Dupuy. Assistances numériques domiciliaires pour les personnes âgées fragiles : Etudes de conception et d'évaluation pilote d'une technologie ambiante d'assistance domiciliaire basée sur l'orchestration d'objets connectés., 2021.

[27] André Neveu. Nourrir le monde en 2050 : un défi qui reste bien réel. *Paysans societe*, 387(3) :35–41, 2021.

[28] Martine Fournier. Le livre blanc du smart retailing-la transformation phygitale : vers le meilleur des deux mondes. 2021.

[29] Jérôme Chenal, Chiara Ciriminna, Rémi Jaligot, Karine Ginisty, and Florian Rudaz. L'utilisation du numérique dans le contexte des villes de l'afrique de l'ouest. Technical report, EPFL, 2021.

[30] Hicham Mohamed SEKKIL and Mahmoud MEBROUKI. *L'utilisation de l'IA dans les systèmes embarqués pour le développement des options automatiques des voitures intelligentes.* PhD thesis, Directeur : M. MEGNAFI Hichem/Co-Directeur : Melle. Imane NEDAJR, 2021.

[31] Aurélien DUMAINE. Évolution d'internet : confiance, identités, vie privée, statut des données et modèles économiques. 2012.

[32] Malcolm Bourdon. *Détection d'intrusion basée sur l'analyse de compteurs matériels pour des objets connectés*. PhD thesis, Toulouse, INSA, 2021.

[33] Youcef OULD YAHIA and Pierre PARADINAS. Applications e-santé, le contrôle des données personnelles un enjeu majeur pour la protection de la vie privée.

[34] Futura. Définition : Sqlite : Futura tech, Dec 2006. consulté le 13/10/2022.