

**Rapport de stage d’été obligatoire**

**Niveau : 1ère année Cycle Ingénieur**

**Spécialité : Génie Informatique Industrielle**

**Réalisé par :**

**Jed Aroua**

***Développement et réalisation d’un prototype « Smart Agriculture » contrôlable via interface web***

**Entreprise d’accueil :**

****

**Encadrant professionnel : Jrad Kais**

**Remerciements**

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.  
  
Tout d'abord, j'adresse mes remerciements à mon encadrant, **Jrad Kais** gérant de la société **MEGADEV** Je remercie également toute l'équipe de WG consulting pour leur accueil, leur esprit d'équipe qui m’ont fait profiter de leurs expériences et m’ont procuré une ambiance propice au bon déroulement de travail. et en particulier mon encadrant **Jrad Kais**, qui m'a beaucoup aidé à comprendre les fondamentaux de développement web .  
Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage

**Table des matières**

**INTRODUCTION** **1**

**1 PRESENTATION DE LA SOCIETE1**

1.1 PRESENTATION GENERALE2

1.2 LES SERVICES3

**2 Etude de l’existant4**

1.1 Problématique5

1.2 Solution6

**3 TECHNOLOGIES UTULISEES4**

3.1 BOOTSTRAP5

3.2 ESP326

3.2.1 Utilité de l ESP326

3.2.2 Connectivité6

3.2.3 Installation6

**4 Langages utilisé4**

4.1 HTML5

4.2 CSS6

4.3 JAVASCRIPT6

4.4 C /C++6

**5 PARTIE WEB 1**

5.1 Présentation de la parité web 2

5.2 Code solution3

**6 PARITE ELECTRONIQUE4**

6.1 Matériel utilisé 5

6.2 Schéma électrique67

6.3 Code solution6

**CONCLUSION GENERALE** **1**

**Introduction**

A mesure que le monde qui nous entoure tend à être toujours plus connecté, beaucoup de nouveaux systèmes intelligents apparaissent. Cette révolution du numérique se traduit par la mise à disposition auprès du particulier de plusieurs outils connectés. Le principal objectif de ces objets est d’obtenir l’ensemble des informations que l’utilisateur cherche en quelques secondes seulement. On peut aussi relever les nombreux progrès de la domotique ces dernières années.

Parlons domotique! Le principe peut être très simple. Dès lors que l’on trouve le moyen de piloter électriquement un appareil, par un relais par exemple pour les systèmes fonctionnant en tout ou rien, il suffit ensuite de piloter ce relais avec une carte connectée au WiFi domestique; on peut alors contrôler l’activation de l’objet depuis n’importe quel endroit sur terre à condition d’avoir un accès à internet. L’idée est ici de séparer la partie puissance de la partie intelligence/commande. Libre à l’utilisateur ensuite d’ajouter un capteur de présence dans une pièce ou un capteur de luminosité par exemple.

De nombreuses solutions coûteuses et “sécurisées” sont aujourd’hui sur le marché, mais il est tout à fait possible de réaliser ce type d’objet pour quelques euros et un peu de temps personnel.

Cette série de tutoriels sur l’ESP32 va vous familiariser avec l’«IoT» (*Internet of Things*), qui vise à acquérir des données et commander différents objets à distance notamment, en ajoutant de l’intelligence au système.

**1 Présentation de la société d’accueil :**

* 1. présentation générale:

**Mega-DEV** est une agence numérique spécialisée dans la création des sites Web personnalisés, développement des applications mobile et le marketing digital qui contribuent à des conversions plus élevées, assurons la réussite du projet en fournissant un environnement de travail d’équipe sur chaque projet et un flux de travail rapide pour offrir un résultat de qualité dans des délais rapides.

1.2 services :

#### **DÉVELOPPEMENT web:**

**Mega-DEV** est un acteur dynamique dans les métiers du conseil, du développement et des services dans l’ingénierie informatique en général. Leurs mission consiste à ont utilisons des technologies sophistiquées et des pratiques professionnelles créer des sites Internet personnalisés  qui augmentent l’engagement des utilisateurs. accompagner les clients et partenaires pour la mise en place de solutions logicielles et informatiques performantes et innovantes dans les domaines de la finance, de l’industrie …

#### **CONSEIL INFORMATIQUE ET GESTION DE PROJETS:**

**Mega-DEV** offre une gamme complète de services de développement de logiciels personnalisés pour une grande variété de domaines verticaux et commerciaux..

#### **COMMUNICATION POUR UN IT CONSULTING PERSONNALISÉ:**

marketing numérique repose sur la base de la conduite des résultats mesurables. **Mega-DEV** prenent une approche du marketing en ligne en mettant l’accent sur l’augmentation des conversions.

**2 ETUDE DE L’EXISTANT** :

*2.1 PROBLEMATIQUE*

Avec l’adoption croissante de l’Internet des Objets (IoT), les appareils connectés ont pénétré tous les aspects de notre vie, de la santé à la domotique, en passant par l’automobile et les villes intelligentes. Il est donc logique que l’IoT trouve son application dans l’agriculture et, en tant que tel, en améliore considérablement presque toutes les facettes. Pourquoi intégrer l’IoT dans votre exploitation ? Pour quelles applications ? Quels défis l’agriculture intelligente doit-elle relever ?

*2.2 SOLUTION*

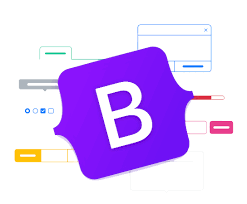
Dans le monde agricole, les solutions IoT prennent la forme de **capteurs reliés à Internet pour collecter des mesures environnementales et mécaniques**. Leur déploiement permet aux agriculteurs de prendre des **décisions éclairées** et améliore presque tous les aspects de leur travail, de l’élevage à l’agriculture.

Le secteur a connu un certain nombre de transformations technologiques au cours des dernières décennies, devenant de plus en plus industrialisé et axé sur la technologie. En utilisant divers outils agricoles intelligents, les agriculteurs ont ainsi acquis un **meilleur contrôle sur le processus d’élevage et de culture**, ce qui le rend plus prévisible et améliore son efficacité.

# **3 technologies utilisées :**

3.1 *Bootstrap* :

J’utilise la framework bootstrap pour la mise en style de mes views (blades) .

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs

# 3.2 Présentation de l’ESP32

L’ESP32 est une petite carte électronique, appelée microcontrôleur, facile à prendre en main grâce à ses ressemblances avec la carte Arduino qui est bien plus répandue. L’ESP32 peut avoir différentes tailles mais la version la plus courante est la suivante :



FIGURE 1 – La carte ESP32 DEVKIT WROOM

## 3.2.1 Utilité de l’ESP32

l’ESP32 est une carte électronique permettant de réaliser des projets «IoT» assez facilement. Elle possède en effet une connectivité assez complète, que nous détaillerons dans la partie suivante. L’ESP32 est assez simple à prendre en main car elle est cousine de la carte Arduino. Il est donc possible de l’utiliser comme la carte microcontrôleur italienne, en utilisant le même langage de programmation, les mêmes modules complémentaires et surtout le même logiciel de compilation (Arduino IDE).

Pas de changement fondamental d’interface pour l’utilisateur, ce qui la rend d’autant plus simple à utiliser.

Par ailleurs, on peut relever que sa taille miniature lui est un avantage considérable : elle mesure en effet moins de 3 cm par 5 cm, malgré les technologies qu’elle embarque! L’ESP32 est sans doute la carte microcontrôleur qui possède le meilleur compromis taille / connectivité / entrées sorties. Cela en fait un outil redoutable pour la miniaturisation des projets connectés.

Pour finir elle est aussi tout à fait adaptée à la réalisation de prototypes et non uniquement à la réalisation de projets «finis et définitifs». Elle est une référence dans l’univers maker pour ces nombreuses raisons, il est donc très facile de trouver de l’aide sur Internet pour utiliser cette carte.

## 3.2.2 Connectivité

L’objet «IoT» est très intéressant, cependant il nécessite un accès à Internet afin de pouvoir assurer une communication permanente avec l’utilisateur. L’ESP32 est en ce sens une solution intéressante car elle contient au sein de son architecture un module WiFi intégré. Dans cette mesure, et donc sans avoir besoin de monopoliser des broches en ajoutant un module WiFi externe, l’utilisateur va être capable de connecter son objet IoT à Internet grâce à trois lignes de codes (pour les réseaux domestiques classiques).

Un second atout majeur de l’ESP32 est sa puce Bluetooth intégrée, qui peut aussi être utilisée en mode «BLE» (*Bluetooth Low Energy*) consommant moins d’énergie. La consommation d’énergie est en effet un critère important dans la conception d’objets IoT. Ils requièrent une connexion quasi permanente à Internet et doivent donc rester allumés ou en état de veille à chaque instant afin de pouvoir recevoir les instructions de l’utilisateur. Pour des raisons écologiques et économiques il est ainsi fondamental de réaliser un objet qui consomme le moins d’énergie que possible en état de veille, d’où l’utilité de la technologie BLE notamment.

Côté entrées et sorties, l’ESP32 est semblable aux cartes Arduino et aux cartes Raspberry. On relève en effet l’utilisation des ports GPIO (*General Purpose Input/Output*) qui sont les broches permettant de servir soit d’entrée, soit de sortie selon le programme télévisé.

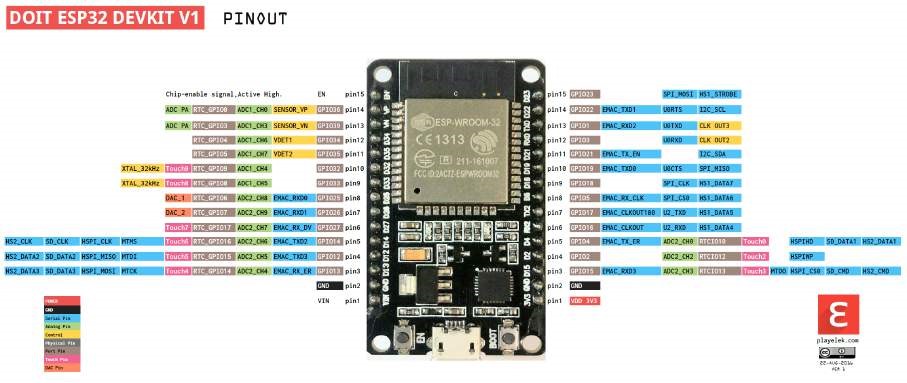


FIGURE 2 – Broches de l’ESP32 DEVKIT

DATA SHEET

………………………………………………………

## 3.2.3 Installation

Passons maintenant à l’étape la plus importante : comment programmer une carte ESP32 avec l’**IDE Arduino**?

Il existe en effet plusieurs méthodes pour programmer une carte ESP32, mais par soucis d’uniformisation et pour limiter le nombre de logiciels requis dans les tutoriels FabLab, nous utiliserons l’outil IDE Arduino qui est assez simple d’utilisation.

La carte Arduino est très facilement programmable avec ce logiciel, cependant il sera nécessaire de **réaliser quelques préparatifs pour pouvoir programmer l’ESP32, notamment installer la carte ESP32 en elle-même**.

**SUIVRE LES ÉTAPES CI-DESSOUS**

1. **Télécharger et installer l’IDE Arduino** depuis le site officiel [Arduino.cc](https://www.arduino.cc/en/Main/Software)

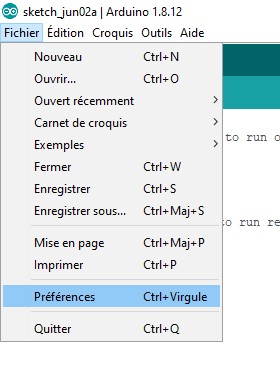


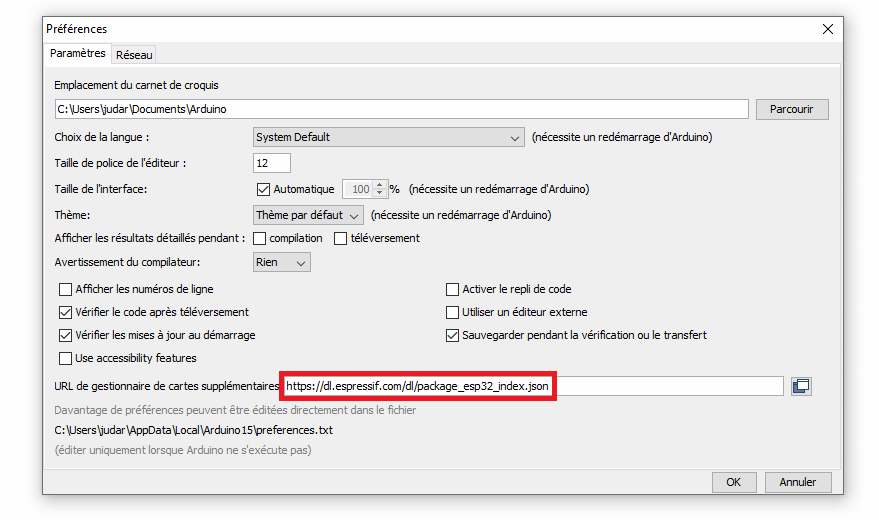
1. **Ajouter la carte ESP32 dans la base des cartes Arduino de l’IDE**

Dans l’IDE Arduino :

* + Ouvrir l’onglet *Fichier > Préférences*
  + Entrer l’adresse suivante dans le champ

*URL de gestionnaire de cartes supplémentaires* : [**ht t ps : //dl . es pr es s i f . c om/dl /pac kage\_es p32\_i ndex. j s on**](https://dl.espressif.com/dl/package_esp32_index.json)





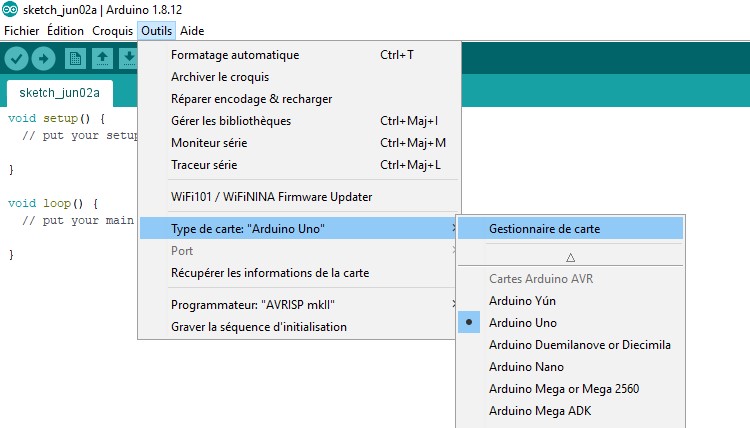
1. **Installer la carte ESP32 dans l’IDE Arduino**

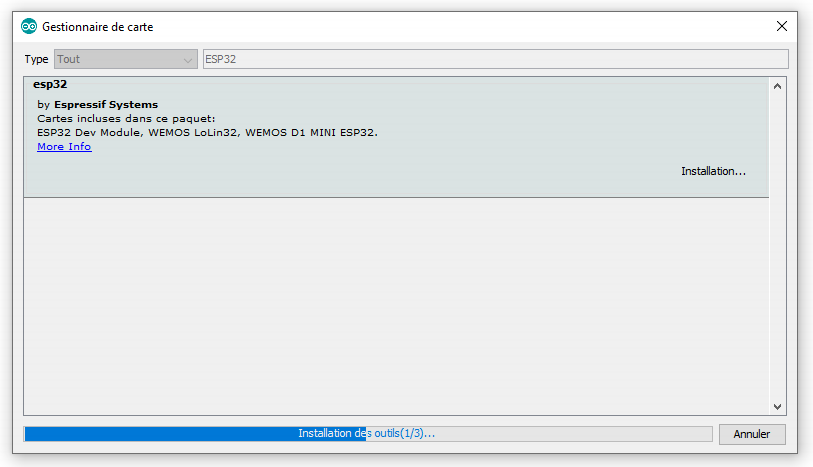
Toujours dans l’IDE Arduino :

* Ouvrir l’onglet

*Outil > Type de carte > Gestionnaire de carte*

* Rechercher le paquet *ESP32*
* Installer le paquet : sélectionnez la dernière version disponible et cliquer sur *Installer*





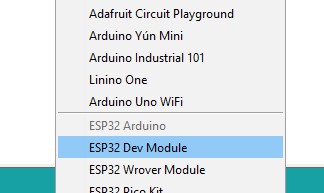
**4. Sélectionner la carte ESP32**

Il ne reste plus qu’à définir la carte ESP32 pour que le programme soit correctement envoyé sur la carte. Toujours dans l’IDE Arduino :

* Ouvrir l’onglet

*Outil > Type de carte*

* Choisissez alors la carte *ESP32* qui correspond à votre version d’ESP32. Le plus souvent ce sera la carte ***ESP32 Dev Module***

Avant de passer au serveur Web, vous devez connecter le capteur DHT11 ou DHT22 à l'ESP32, comme indiqué dans le schéma ci dessus.

Dans ce cas, nous connectons la broche de données au GPIO 27, mais vous pouvez la connecter à n'importe quelle autre broche numérique.

**5.Installation des bibliothèques nécessaires :**

Il faut installer quelques bibliothèques pour ce projet :

-Les bibliothèques DHT et Adafruit Unified Sensor Driver à lire à partir du capteur DHT.

-Bibliothèques ESPAsyncWebServer et Async TCP pour créer le serveur Web asynchrone.

-Serveur Web asynchrone ESP32 avec le DHT11 ou DHT22 qui affiche la température et l'humidité à l'aide d'Arduino IDE.

-Le serveur Web que nous allons construire met à jour les lectures automatiquement sans avoir besoin d'actualiser la page Web.

-Pour créer le serveur Web, nous utiliserons la bibliothèque ESPAsyncWebServer qui fournit un moyen simple de créer un serveur Web asynchrone. Construire un serveur web asynchrone présente plusieurs avantages, tels que :

« Gérer plus d'une connexion en même temps » ;

« Lorsque vous envoyez la réponse, vous êtes immédiatement prêt à gérer d'autres connexions pendant que le serveur se charge d'envoyer la réponse en arrière-plan » ;

« Moteur de traitement de modèles simple pour gérer les modèles » ;

3.3 Github :

Github est une entreprise de développement logiciel et de service dont le siège est situé aux États-Unis. Github développe notamment la plateforme Github, l'éditeur de texte Atom ou encore le framework Electron.

**4 Langages de programmations :**

|  |  |
| --- | --- |
| **Langages** | **rôle** |
| Html  CSS  Java script | frontend |
| C/C++ | PROGRAMATION DE LA CARTE ARDUINO |

4.1 HTML:

Le ***HyperText Markup Language***, généralement abrégé **HTML** ou dans sa dernière version [**HTML5**](https://fr.wikipedia.org/wiki/HTML5), est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web).

Ce langage permet :

* d’écrire de l’[hypertexte](https://fr.wikipedia.org/wiki/Hypertexte), d’où son nom,
* de structurer [sémantiquement](https://fr.wikipedia.org/wiki/S%C3%A9mantique) la page,
* de mettre en forme le contenu,
* de créer des formulaires de saisie,
* d’inclure des [ressources](https://fr.wikipedia.org/wiki/Ressource_du_World_Wide_Web) [multimédias](https://fr.wikipedia.org/wiki/Multim%C3%A9dia) dont des [images](https://fr.wikipedia.org/wiki/Image_num%C3%A9rique), des [vidéos](https://fr.wikipedia.org/wiki/Vid%C3%A9o), et des programmes informatiques,
* de créer des documents [interopérables](https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique) avec des équipements très variés de manière conforme aux exigences de l’[accessibilité du web](https://fr.wikipedia.org/wiki/Accessibilit%C3%A9_du_web).



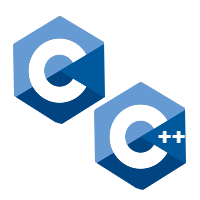
4.2 CSS :

Les **feuilles de style en cascade**, généralement appelées **CSS** de l'anglais ***Cascading Style Sheets***, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

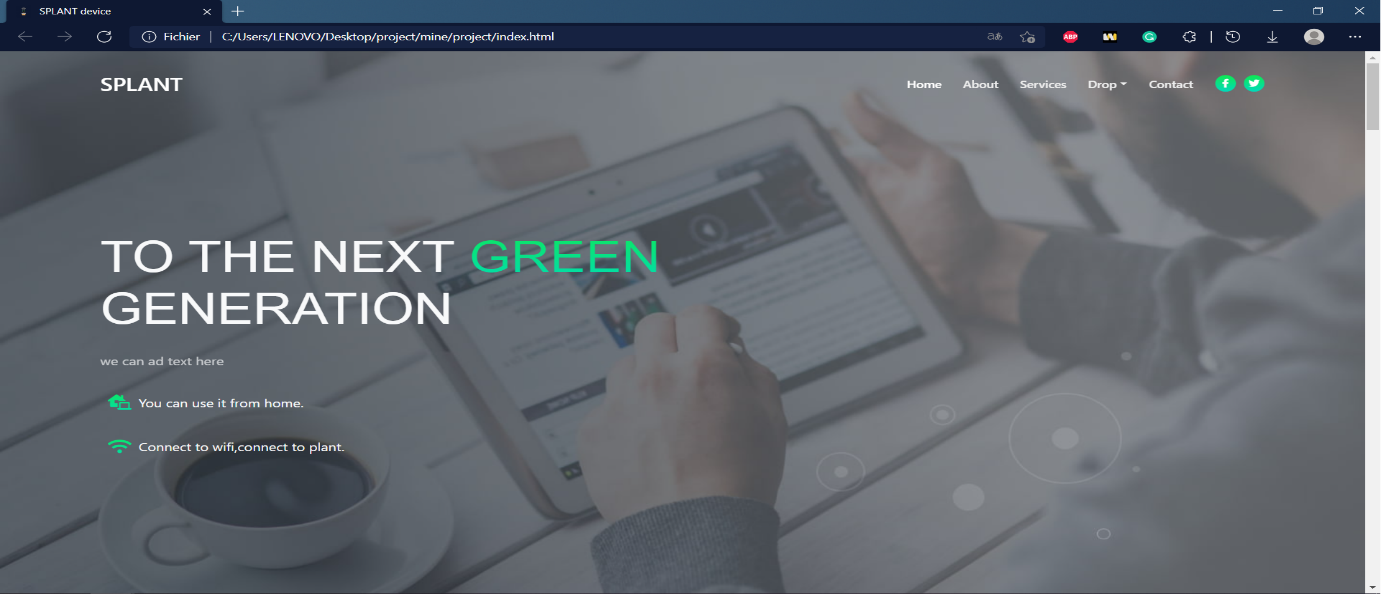
4.3 Java Script :

**JavaScript** est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) de [scripts](https://fr.wikipedia.org/wiki/Langage_de_script) principalement employé dans les [pages web](https://fr.wikipedia.org/wiki/Pages_web) interactives et à ce titre est une partie essentielle des [applications web](https://fr.wikipedia.org/wiki/Application_web). Avec les [technologies](https://fr.wikipedia.org/wiki/Technologie) [HTML](https://fr.wikipedia.org/wiki/HTML) et [CSS](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade), JavaScript est parfois considéré comme l'une des [technologies](https://fr.wikipedia.org/wiki/Technologie) cœur du [World Wide Web](https://fr.wikipedia.org/wiki/World_Wide_Web). Une grande majorité des [sites web](https://fr.wikipedia.org/wiki/Site_web) l'utilisent, et la majorité des [navigateurs web](https://fr.wikipedia.org/wiki/Navigateur_web) disposent d'un [moteur JavaScript](https://fr.wikipedia.org/wiki/Moteur_JavaScript) dédié pour l'[interpréter](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)), indépendamment des considérations de sécurité qui peuvent se poser le cas échéant.C'est un langage [orienté objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) à [prototype](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_prototype) : les bases du langage et ses principales [interfaces](https://fr.wikipedia.org/wiki/Interface_de_programmation) sont fournies par des [objets](https://fr.wikipedia.org/wiki/Objet_(informatique)).

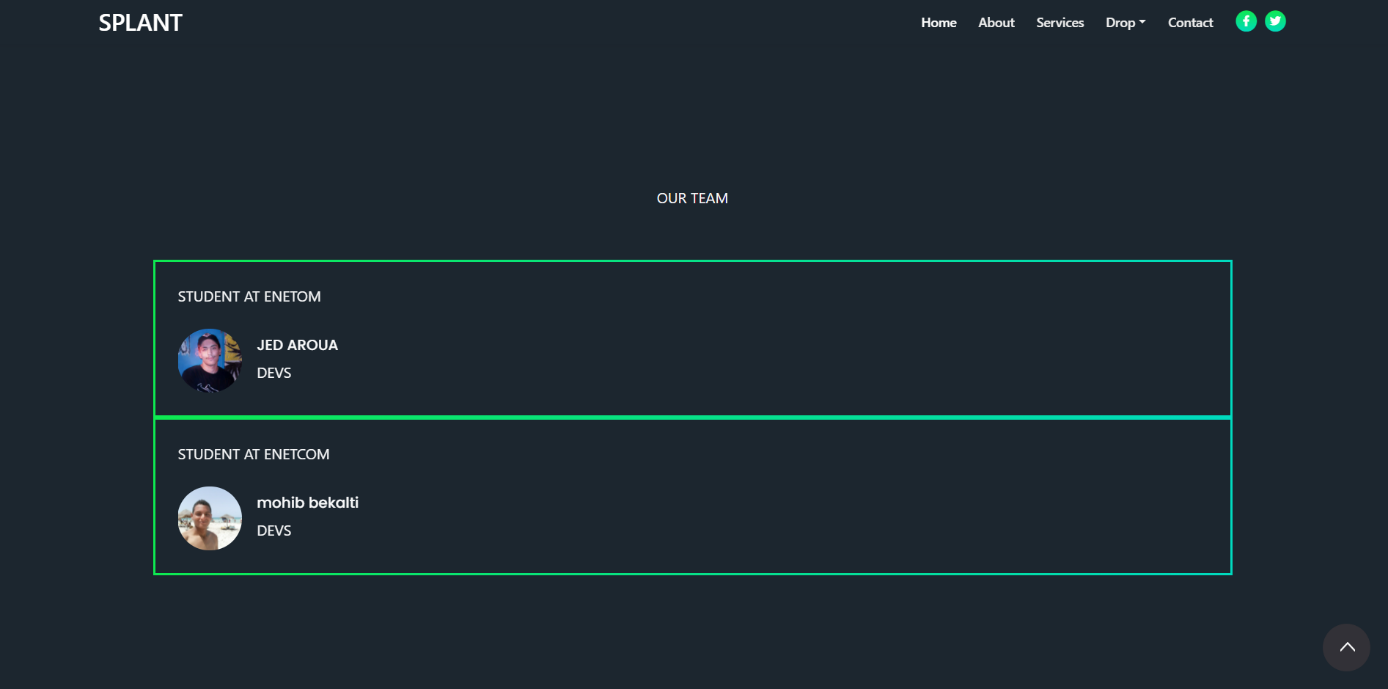
4.4 c/c++ :

**C/C++** est un langage de programmation compilé permettant la programmation sous de multiples paradigmes, dont la programmation procédurale, la programmation orientée objet et la programmation générique. Ses bonnes performances, et sa compatibilité avec le C en font un des langages de programmation les plus utilisés dans les applications où la performance est critique.

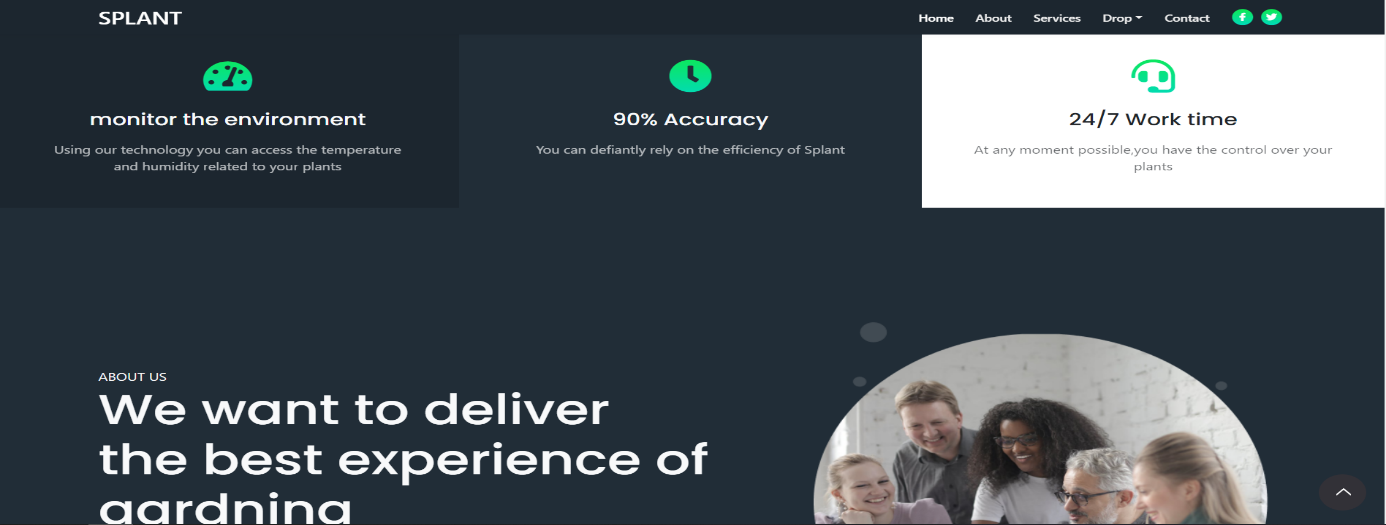
**5 PARTIE WEB :**

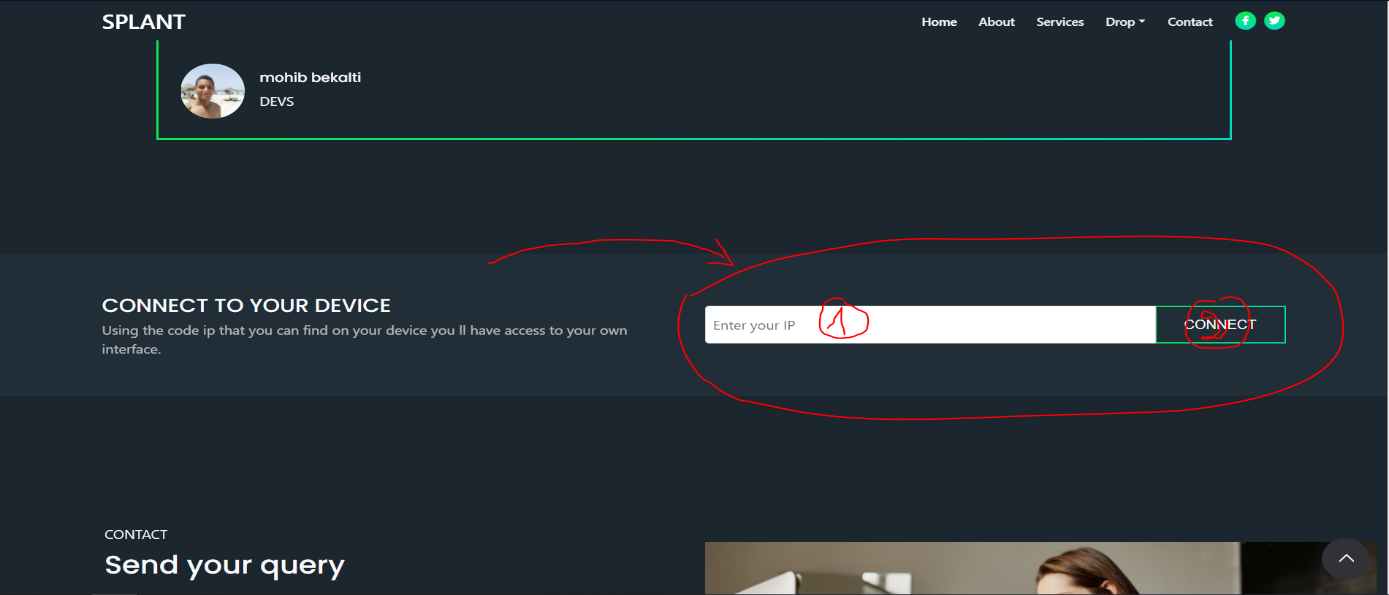
5.1 Présentation de la parité web

-Page d’accueil-



-Page de groupe –

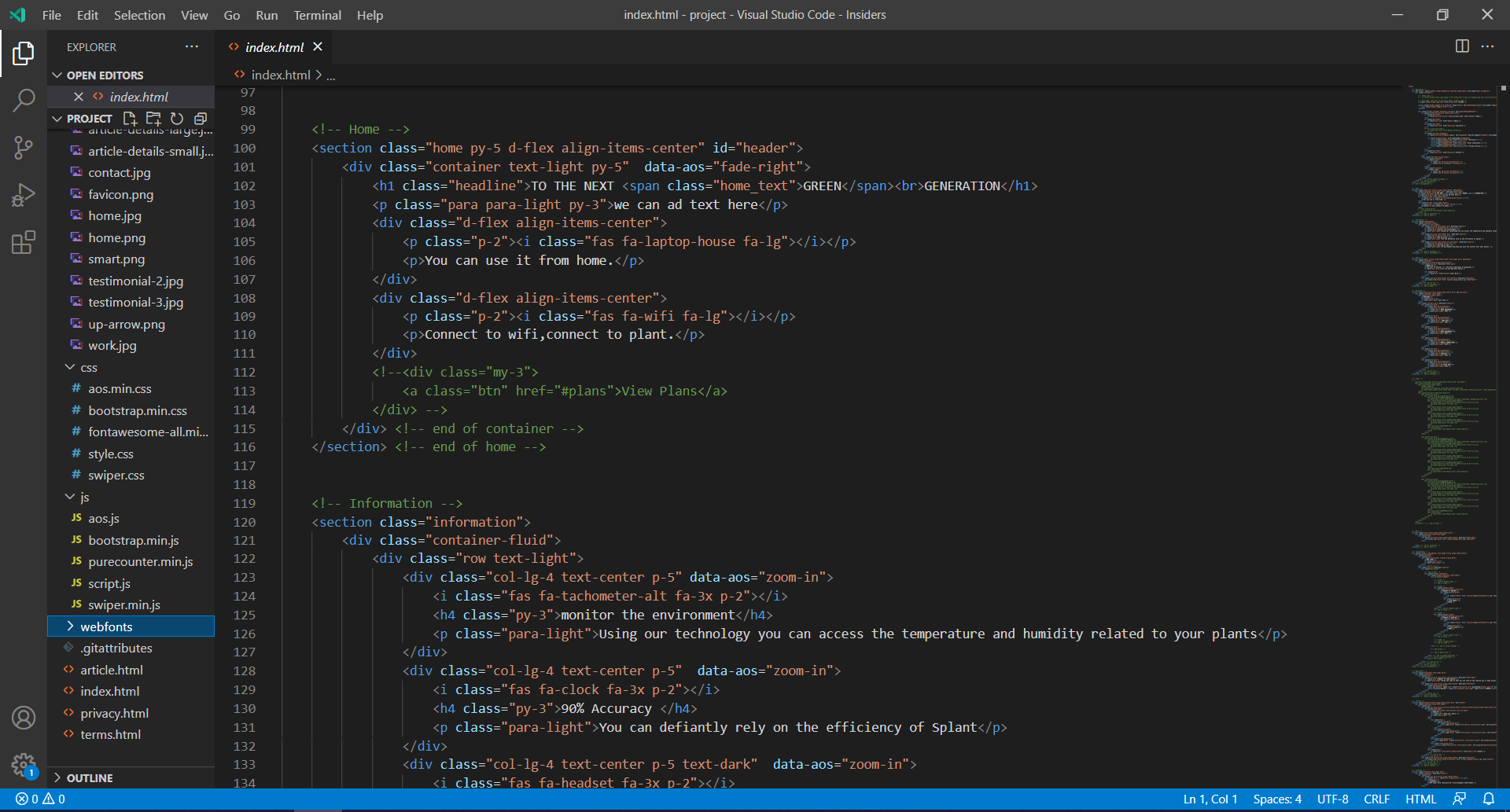
-Page d information-



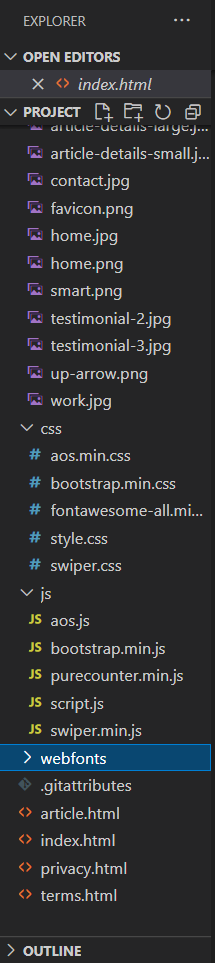
-Page pour connecter-

5.2 code solution

Utilisant Visual Studio, il était facile grâce au débuggueur intégré de réparer les erreurs. De plus, une simple actualisation de la page sur laquelle je travaillais et l’utilisation des fonctionnalités pouvaient me montrer s’il y avait des erreurs. Les tests étaient donc nombreux, presqu’à chaque modification, je testais au cas où. Je pouvais donc remarquer s’il manquait quelque chose ou si je pouvais faire beaucoup mieux. Les erreurs de compilation du site web sont aussi immédiatement indiquées soit dans le logiciel si je lançais l’ouverture de la page dans un navigateur web, soit dans une page d’erreur sur le navigateur lui-même si j’actualisais la page.



-Organisation du code-

-gestion des fichiers-

**6 PARTIE ELECTRONIQUE :**

6.1 MATERILE:

-Carte de développement ESP32

-Capteur de température et d'humidité DHT22 ou DHT11

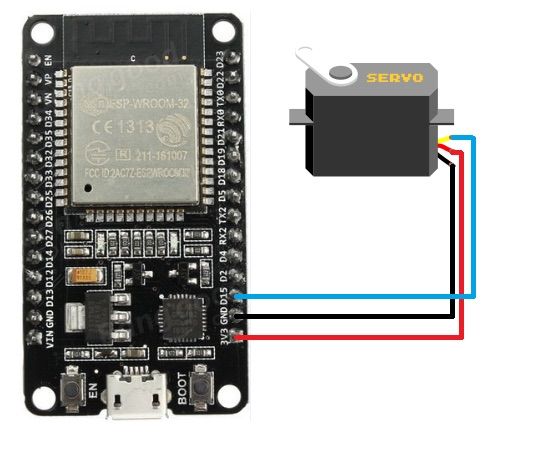
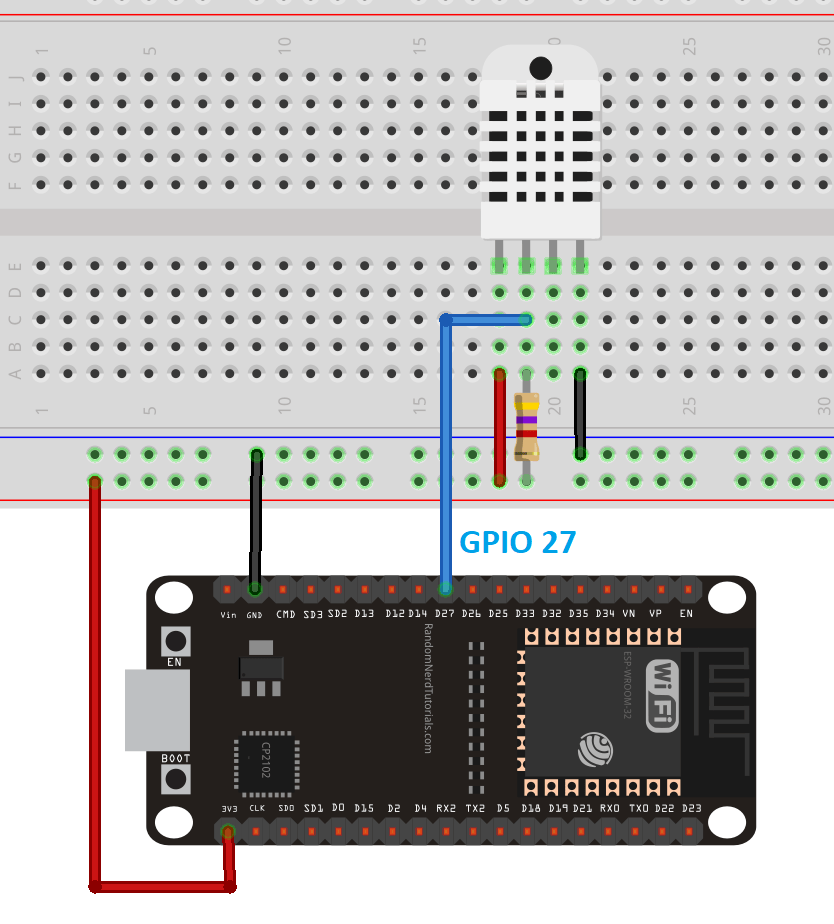
-Résistance 4.7k Ohm

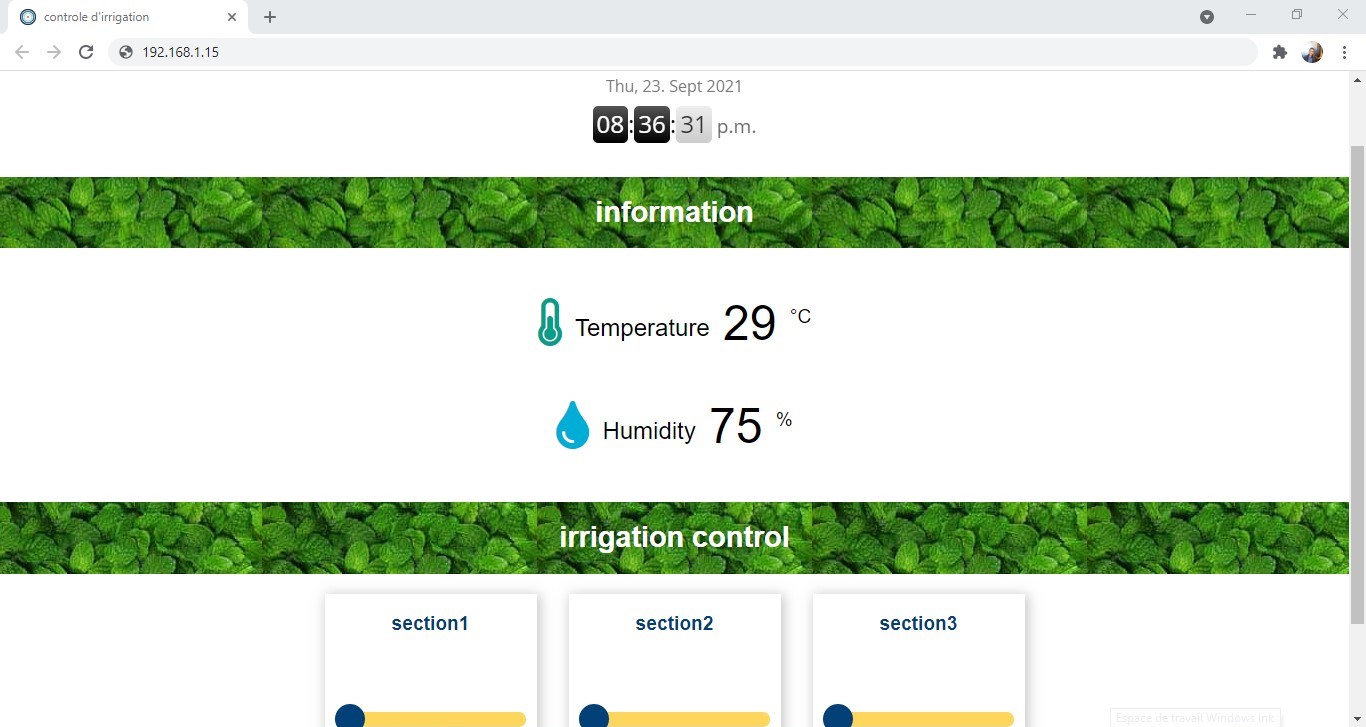
-Planche à pain

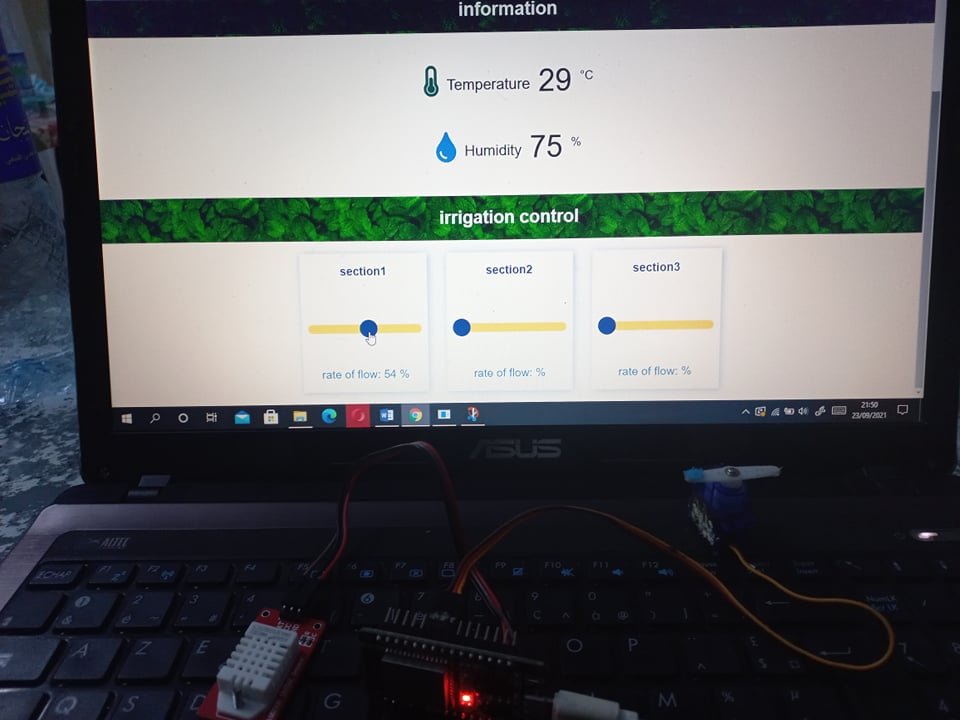
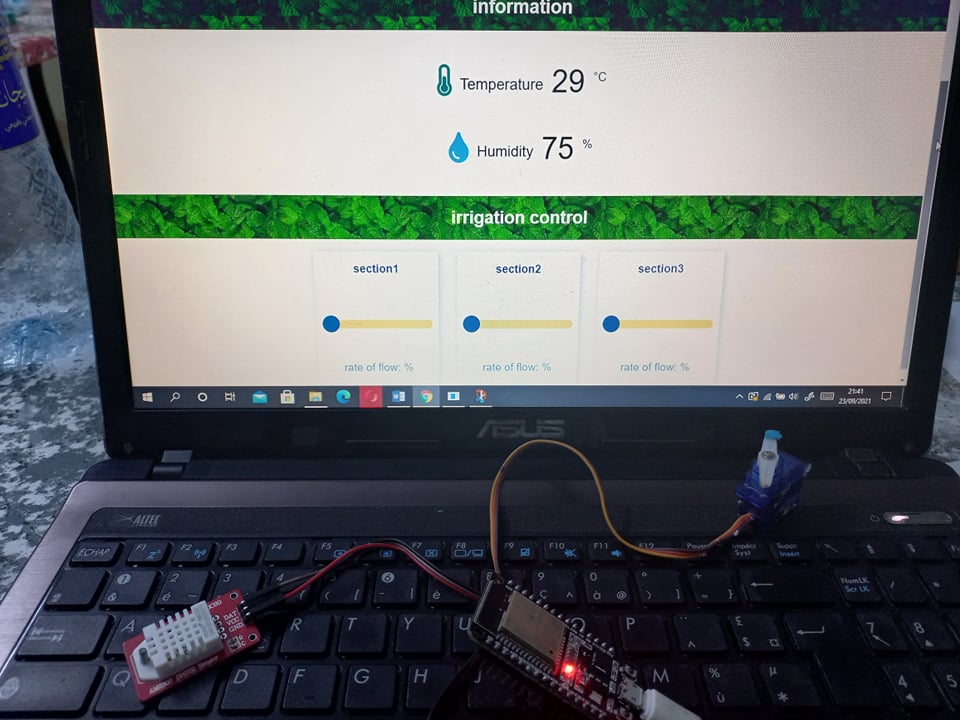
-Fils de cavalier

-servo moteur(pour l’électrovanne)

6.2 SCHEMA ELECTRIQUE:

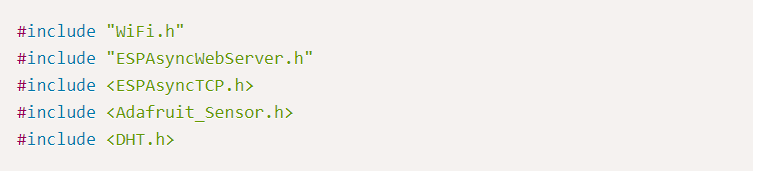
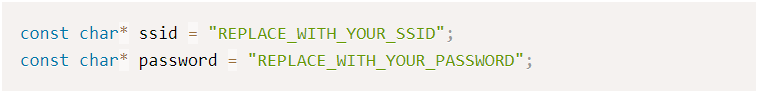


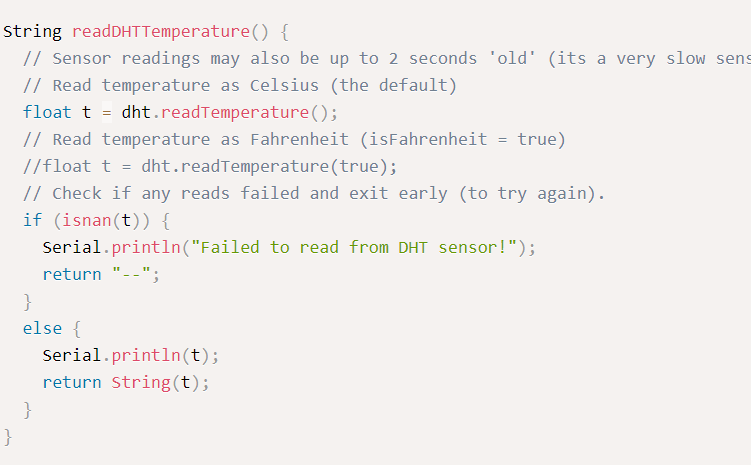




6.3 CODE SOLUTION:

1- température, humidité :

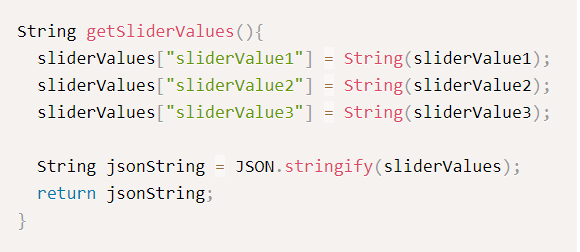
-Importation des bibliothèques : Tout d'abord, importez les bibliothèques requises. Le WiFi, ESPAsyncWebServer et ESPAsyncTCP sont nécessaires pour construire le serveur Web. Les bibliothèques Adafruit\_Sensor et DHT sont nécessaires pour lire à partir des capteurs DHT11 ou DHT22.-Insérez vos identifiants réseau dans les variables suivant : -Définition des variables : Définissez le GPIO auquel la broche de données DHT est connectée. Dans ce cas, il est connecté au GPIO 27. -Ensuite, sélectionnez le type de capteur DHT que vous utilisez. Dans notre exemple, nous utilisons le DHT22. Si vous utilisez un autre type, il vous suffit de décommenter votre capteur et de commenter tous les autres. -Instanciez un objet DHT avec le type et la broche que nous avons définis précédemment.-Créer un objet AsyncWebServer sur le port 80-les fonctions pour lire la température et humidité :Nous avons créé deux fonctions : une pour lire la température (readDHTTemperature()) et l'autre pour lire l'humidité (readDHTHumidity()).

Meme fonction pour lire l’humidite.

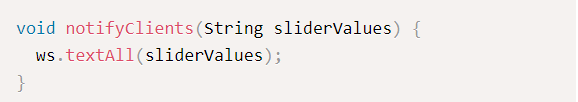
2-Contrôle d’irrigation :

Dans cet partir en s'intéresse à la fonctionnement de la servo moteur qui la compsante principale de l’electrovane qui fait la contrôle de couler d’eau avec une debit variable.

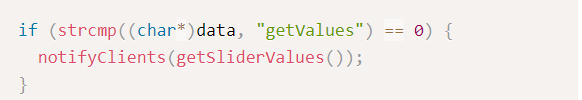
-La fonction getSliderValues() crée une chaîne JSON avec les valeurs actuelles du curseur



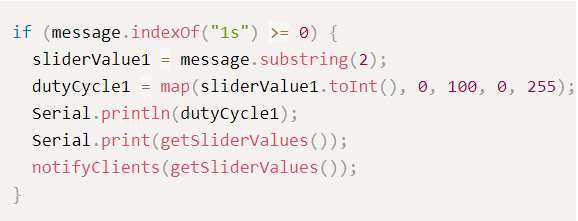
-La fonction notifyClients() notifie à tous les clients les valeurs actuelles du curseur. L'appel de cette fonction est ce qui nous permet de notifier les changements dans tous les clients chaque fois que vous définissez une nouvelle position pour un curseur



- Le handleWebSocketMessage(), comme son nom l'indique, gère ce qui se passe lorsque le serveur reçoit un message du client via le protocole WebSocket. Nous avons vu dans le fichier JavaScript, que le serveur peut recevoir le message getValues ​​ou un message avec le numéro du slider et la valeur du slider. Lorsqu'il reçoit le message getValues, il envoie les valeurs actuelles du curseur.



S'il reçoit un autre message, nous vérifions à quel curseur correspond le message et mettons à jour la valeur de cycle de service correspondante. Enfin, nous informons tous les clients qu'un changement s'est produit. Voici un exemple pour le curseur 1 :



Dans la boucle(), nous mettons à jour le rapport cyclique des canaux PWM pour chager le débit de l’eau.

**Conclusion**

- Vécu et expérience-

Ce stage m’a aussi offert la chance de progresser techniquement dans les domaines que je connaissais déjà tels que le Html, le Css, le Javascript ou encore les optimisations de rapidité. J’ai également pu prendre en main Visual Studio avec lequel je suis maintenant familière. Comme je connaissais peu le langage de programmation C/C++, ce stage m’a alors permis de découvrir ce langage que j’apprécie d’ailleurs énormément.

-Difficultés-

De jour en jour, j’améliorais ma façon de penser et de coder le site web avec l’implémentation de l’interface de la carte électronique . Lorsque je revenais sur un code plus ancien, je me rendais compte que ce n’était pas comme je le souhaitais. Parfois même j’y trouvais des erreurs ou des fautes de goût quant à la manière d’avoir implémenter une partie. Cela fait donc partie des difficultés que j’ai rencontrées : un souci de clarté et de compréhension.

-Synthèse du travail-

En conclusion, ce stage m’a permis de faire évoluer une version désuète en un système plus dynamique, attrayant et fonctionnel bien qu’il reste encore quelques bugs. Je suis satisfaite du travail que j’ai fourni et de son résultat. Cependant, toutes mes idées d’amélioration n’ont pu être intégrées par manque de temps.