Université Cadi Ayyad

École Nationale des Sciences Appliquées Safi

Département Informatique

$2^{éme}$ Année Génie Informatique

Année Universitaire : 2023/2024

IoT-based intelligent nature environmental monitoring system

prepared by:

- **KHOUY Abderrazzak**
- **MATRAB Mohamed**

1. Introduction:

In a world increasingly aware of the importance of environmental protection, technological advances offer innovative solutions to monitor and preserve our planet. The "Environmental Serviance" project is a remarkable example of the application of the Internet of Things (IoT) technology and cloud computing to monitor environmental conditions with precision and efficiency.

2. Development:

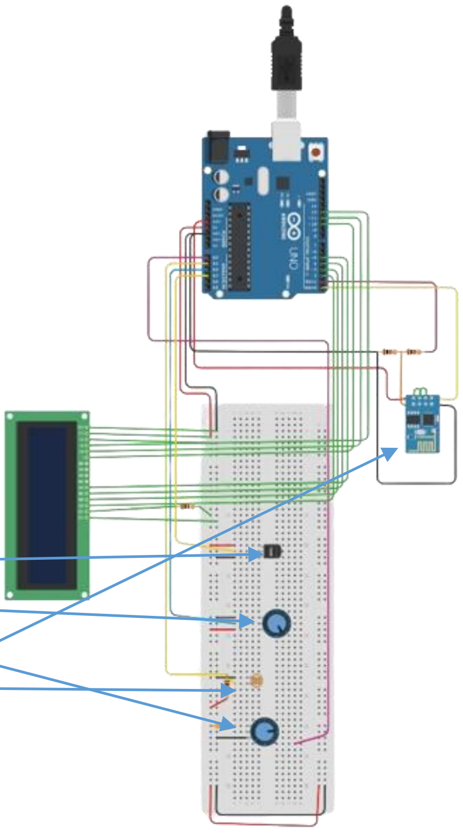
The "Environmental Serviance" project relies on the use of IoT sensors to collect essential environmental data such as:La température.

- Humidity
- Wind speed
- Light intensity

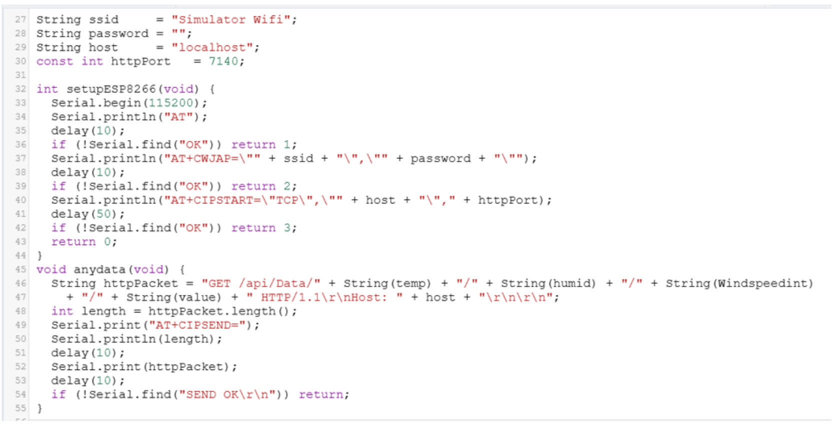
This data is collected in real time using a set of sensors that includes:

- Temperature sensor.
- A potentiometer simulating humidity and wind speed
- Photoresistor simulating a light sensor.

All of this data is aggregated using a TinkerCAD map.



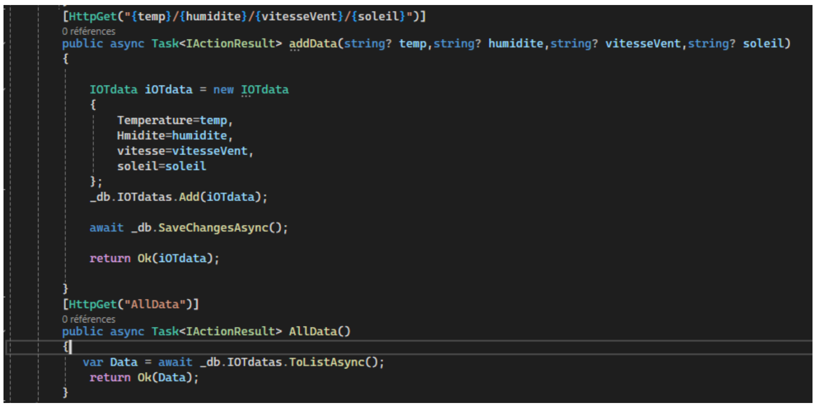
To send the data to the PLC, a WiFi module is ESP8266. This module is responsible for collecting data from the sensors and transmitting it to the PLC via the WiFi network.



```
String ssid     = "Simulator Wifi";
String password = "";
String host     = "localhost";
const int httpPort   = 7140;

int setupESP8266(void) {
  Serial.begin(115200);
  Serial.println("AT");
  delay(10);
  if (!Serial.find("OK")) return 1;
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
  delay(10);
  if (!Serial.find("OK")) return 2;
  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);
  if (!Serial.find("OK")) return 3;
  return 0;
}
void anydata(void) {
  String httpPacket = "GET /api/Data/" + String(temp) + "/" + String(humid) + "/" + String(Windspeedint)
    + "/" + String(value) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length = httpPacket.length();
  Serial.print("AT+CIPSEND=");
  Serial.println(length);
  delay(10);
  Serial.print(httpPacket);
  delay(10);
  if (!Serial.find("SEND OK\r\n")) return;
}
```

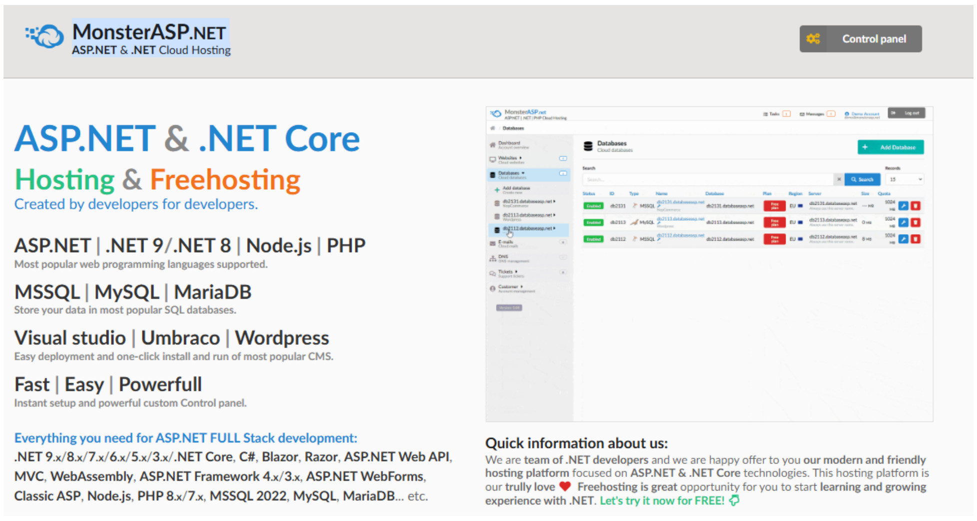
Figure 1 envoyer les donnes en utilisant un module WiFi ESP8266

Once collected, the data is sent to an API developed with ASP.NET Core API. Here's sample code for the API:



```
[HttpGet("{temp}/{humidite}/{vitesseVent}/{soleil}")]
0 références
public async Task<IActionResult> addData(string? temp,string? humidite,string? vitesseVent,string? soleil)
{

    IOTdata iOTdata = new IOTdata
    {
        Temperature=temp,
        Hmidite=humidite,
        vitesse=vitesseVent,
        soleil=soleil
    };
    _db.IOTdatas.Add(iOTdata);

    await _db.SaveChangesAsync();

    return Ok(iOTdata);

}
[HttpGet("AllData")]
0 références
public async Task<IActionResult> AllData()
{
   var Data = await _db.IOTdatas.ToListAsync();
    return Ok(Data);
}
```

This API is hosted in the cloud, specifically on the MonsterASP.NET ASP.NET & .NET Cloud Hosting platform, providing maximum accessibility and availability. The use of the cloud allows for secure and scalable storage of data, ensuring its integrity and availability at all times.



MonsterASP.NET
ASP.NET & .NET Cloud Hosting

Control panel

ASP.NET & .NET Core
Hosting & Freehosting
Created by developers for developers.

ASP.NET | .NET 9/.NET 8 | Node.js | PHP
Most popular web programming languages supported.

MSSQL | MySQL | MariaDB
Store your data in most popular SQL databases.

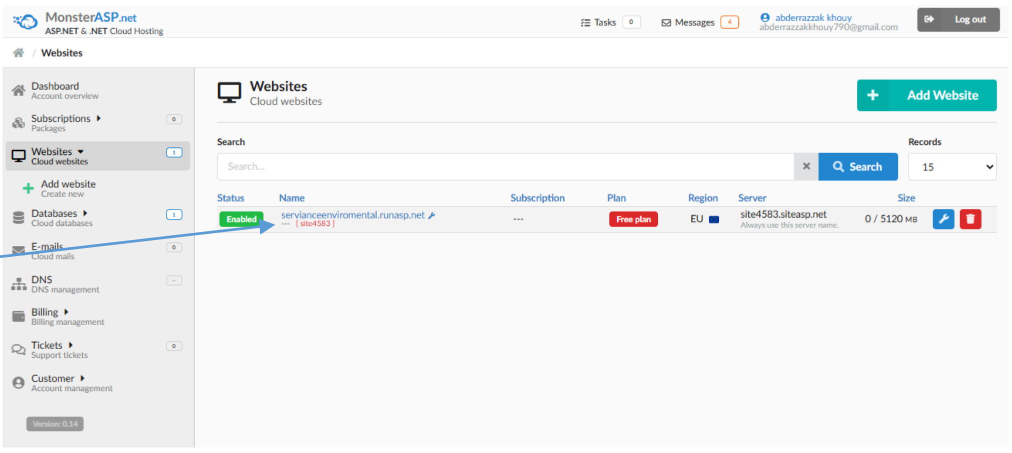
Visual studio | Umbraco | Wordpress
Easy deployment and one-click install and run of most popular CMS.

Fast | Easy | Powerfull
Instant setup and powerful custom Control panel.

Everything you need for ASP.NET FULL Stack development:
.NET 9.x/8.x/7.x/6.x/5.x/3.x/.NET Core, C#, Blazor, Razor, ASP.NET Web API, MVC, WebAssembly, ASP.NET Framework 4.x/3.x, ASP.NET WebForms, Classic ASP, Node.js, PHP 8.x/7.x, MSSQL 2022, MySQL, MariaDB... etc.

Quick information about us:
We are team of .NET developers and we are happy offer to you our modern and friendly hosting platform focused on ASP.NET & .NET Core technologies. This hosting platform is our trully love ❤ Freehosting is great opportunity for you to start learning and growing experience with .NET. Let's try it now for FREE!

The creation of a site (Environmental Service) to host the API.

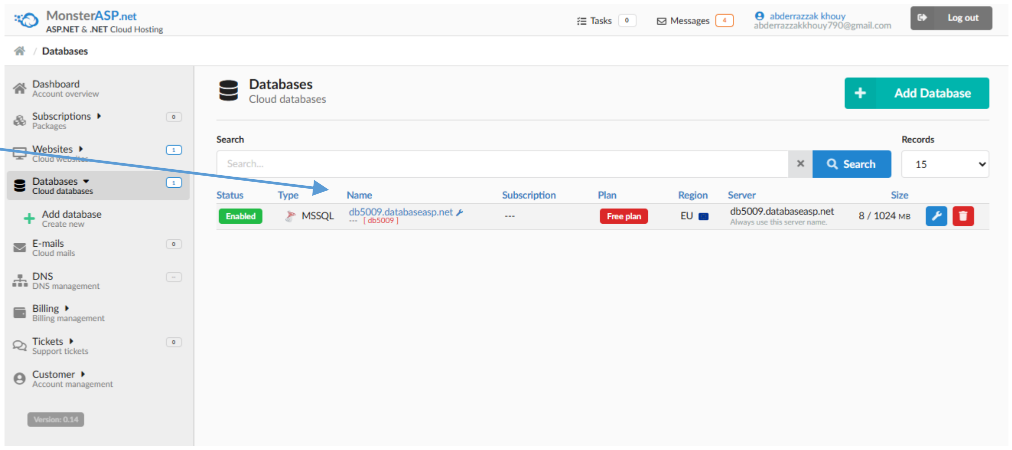


MonsterASP.net
ASP.NET & .NET Cloud Hosting
Tasks 0
Messages 4
abderrazzak khouy
abderrazzakkhouy790@gmail.com
Log out
Websites
Dashboard Account overview
Subscriptions Packages 0
Websites Cloud websites 1
Add website Create new
Databases Cloud databases 1
E-mails Cloud mails 0
DNS DNS management
Billing Billing management
Tickets Support tickets 0
Customer Account management
Version: 0.14
Add Website
Search
Records
15

Websites
Cloud websites

Status	Name	Subscription	Plan	Region	Server	Size
Enabled	servianceenviromental.runasp.net --- [site4583]	---	Free plan	EU	site4583.siteasp.net Always use this server name.	0 / 5120 MB

Creating a database to store the data sent by the WiFi module ESP8266.

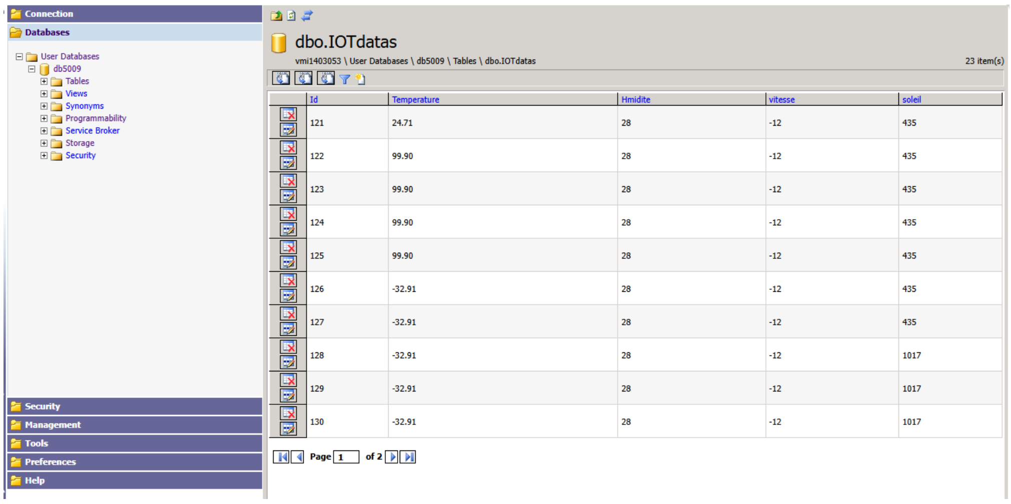


MonsterASP.net
ASP.NET & .NET Cloud Hosting
Tasks 0
Messages 4
abderrazzak khouy
abderrazzakkhouy790@gmail.com
Log out
Databases
Dashboard Account overview
Subscriptions Packages 0
Websites Cloud websites 1
Databases Cloud databases 1
Add database Create new
E-mails Cloud mails 0
DNS DNS management
Billing Billing management
Tickets Support tickets 0
Customer Account management
Version: 0.14
Add Database
Search
Records
15

Databases
Cloud databases

Status	Type	Name	Subscription	Plan	Region	Server	Size
Enabled	MSSQL	db5009.databaseasp.net --- [db5009]	---	Free plan	EU	db5009.databaseasp.net Always use this server name.	8 / 1024 MB

Once the data is sent to the API, the API stores it in the database.



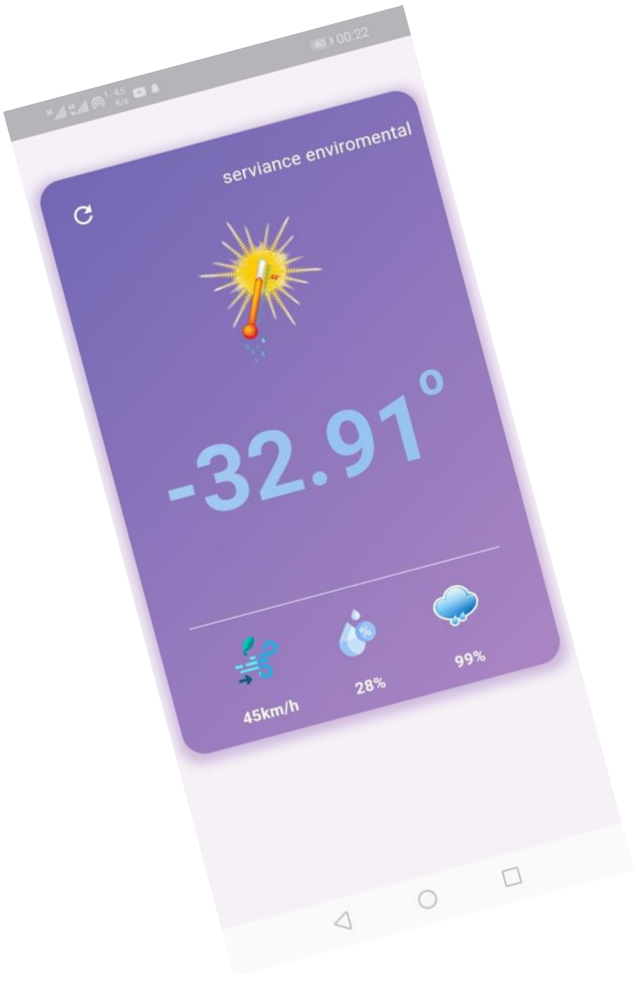
dbo.IOTdatas

vmi1403053 \ User Databases \ db5009 \ Tables \ dbo.IOTdatas 23 item(s)

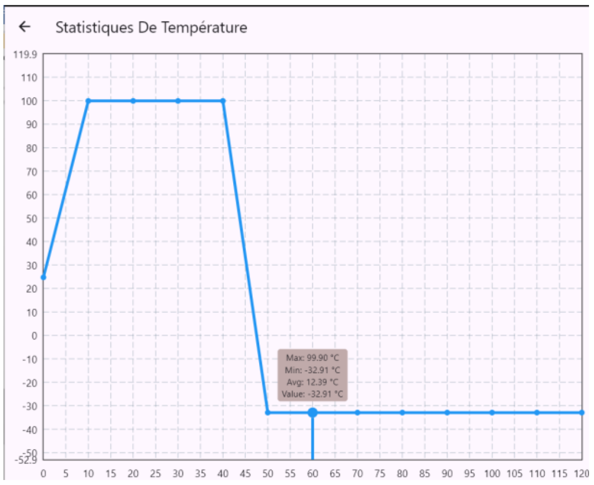
Id	Temperature	Hmidite	vitesse	soleil
121	24.71	28	-12	435
122	99.90	28	-12	435
123	99.90	28	-12	435
124	99.90	28	-12	435
125	99.90	28	-12	435
126	-32.91	28	-12	435
127	-32.91	28	-12	435
128	-32.91	28	-12	1017
129	-32.91	28	-12	1017
130	-32.91	28	-12	1017

Page 1 of 2

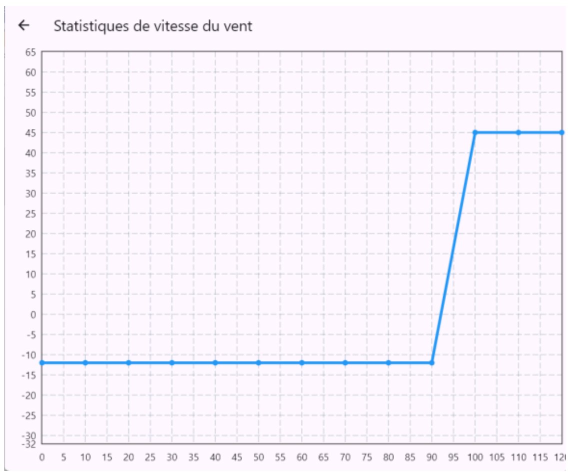
Once the data is stored in the database via the API, it can be accessed and displayed in the mobile application developed with Flutter. The Flutter app interacts with the API to retrieve the latest recorded environmental data. This data is then presented in an intuitive and visually pleasing way within the app, allowing users to view real-time information on temperature, humidity, wind speed and percentage rain prediction. By displaying the latest recorded data, the app ensures that the information presented is always up to date, providing accurate and current environmental monitoring directly from their smartphone or tablet.

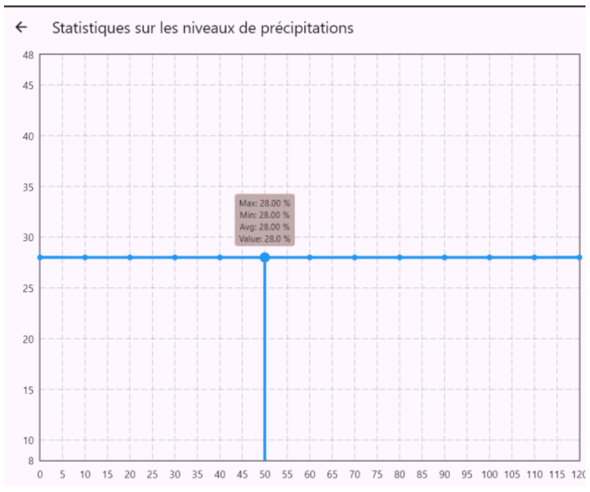


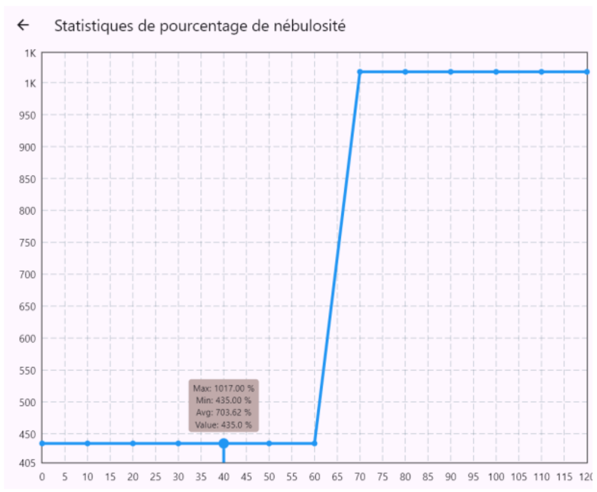
This graph displays historical temperatures as a curve. Detailed information about each point (maximum, minimum, average temperature and exact value) is displayed by hovering over it.



In addition to temperature statistics, the app also offers information on wind speed, precipitation levels and cloud cover percentage.







Conclusion :

The “Environmental Service” project perfectly illustrates the power of IoT technology and cloud computing in the field of environmental monitoring. By combining smart sensors, robust cloud infrastructure and a user-friendly user interface, this project offers a comprehensive and efficient solution for monitoring and understanding environmental conditions in real time.

Although we did not use physical sensors for each environmental parameter, using potentiometers to simulate these sensors proved to be a valuable experiment. This approach validated the functionality of the system and demonstrated the architecture's ability to collect, process and display relevant environmental data. This project remains an excellent demonstration of the possibilities offered by IoT and the cloud, paving the way for more advanced future applications and real-world deployments using specialized sensors. Through such innovations, we are better equipped to protect and preserve our precious environment for future generations.