KOCAELİ ÜNİVERSİTESİ* MÜHENDİSLİK FAKÜLTESİ

IOT TEMELLİ SENSÖR BİLGİLERİNİN WEB ORTAMINDA GÖSTERİMİ VE İZLENMESİ

MÜHENDİSLİK TASARIM-2
ARA RAPORU

Ozan Hamdi Kaplan

Bölüm: Elektronik ve Haberleşme Mühendisliği

Dersi Veren: Doç. Dr.Oğuzhan KARAHAN

KOCAELİ, 2023

ÖNSÖZ ve TEŞEKKÜR

Nesnelerin İnterneti (IOT), internet üzerinden diğer cihazlara ve sistemlere bağlanmak ve bunlarla veri paylaşmak amacıyla sensörler, yazılımlar ve diğer teknolojilerle gömülü fiziksel nesnelerin ağını tanımlar. Bu cihazlar, sıradan ev nesnelerinden gelişmiş endüstriyel araçlara kadar çeşitlilik gösterir.

Bu Mühendislik Tasarımı-2 projesi çalışmasında, NodeJS ve C programlama dillerinin en uygun kodlama yöntemleri kullanılarak IOT ve web uygulama çalışmaları gerçekleştirilmiştir. Çalışmaların bu alanda çalışanlara ve bu alanda çalışacak araştırmacılara faydalı olmasını dilerim.

Değerli hocam Doç. Dr. Oğuzhan Karahan'a sınıf arkadaşlarıma, son sınıf arkadaşlarıma ve mezunlarıma. Özellikle bu süreçte yardımlarını esirgemeyen ailem hem maddi hem de manevi katkılarda bulundu ve katkı sağlamaya devam ediyor. Teşekkürler.

Haziran 2023, KOCAELİ Ozan Hamdi Kaplan

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

KISALTMALAR

IoT	:Internet of Things
GHz	:Gigahertz
ISM	:Industrial Scientific and Medical
MBPS	:Megabit per second
API	:Application Programming İnterface
NET	:Network
FCL	:Full Container Load
RISC	:Reduced instruction set computer
I2C	:Inter-Integrated Circuit
SPI	:Serial peripheral interface
UART	:Universal Asynchronous Receiver Transmitter
USART	:Universal synchronous asynchronous receiver transmitter
ADC	:Analog Digital Converter
LCD	:Liquid-crystal display
IDE	:Integrated development environment
TI	:Texas Instruments
CCS	:Code Composer Studio
IR	:Infrared
PWM	:Pulse Width Modulation
A	:Amper
V	:Volt
W	:Watt
IC	:Integrated Circuit
JS	:JavaScript
DB	:Database
AJAX	: Asynchronous JavaScript and XML

1. GİRİŞ

Bugün aklınıza gelebilecek her nesne (telefonlar, bilgisayarlar, ev sistemleri, robotlar vb.) bir şekilde internete giriyor ve diğer cihazlarla iletişim kuruyor demektir. Ancak Nesnelerin İnterneti, yalnızca İnternet'e bağlanan cihazlardan ibaret değildir. IoT cihazları insanlarla iletişim kurar ve herhangi bir durumdan haberdar olmalarını sağlar. IoT kavramının tarihteki ilk uygulaması, 1991 yılında Cambridge Üniversitesi'nden bir grup akademisyen tarafından kameralı bir sistem sayesinde kahve makinesi görüntülerinin internet üzerinden paylaşılmasından sonra günümüze kadar geliştirilmiştir. Nesnelerin sistemi algılayabilmeleri, iletişim kurabilmeleri, verileri kaydedebilmeleri, hedef adrese yönlendirebilmeleri vb. Yetenekleri sayesinde birbirleriyle iletişim kurabilmeleri aslında uzun süredir hayatımızda.

IoT'nin günümüzde en yaygın uygulama alanlarından biri de neredeyse her gün kullandığımız belediye durakları, metro istasyonları yani Akıllı İstasyonlardır.

NodeMCU ESP32, popüler NodeMCU geliştirme kartının geliştirilmiş bir versiyonudur. ESP32, Espressif Systems tarafından üretilen ve Wi-Fi ve Bluetooth desteği sunan bir mikro denetleyici yongası olan ESP-WROOM-32'yi kullanır. NodeMCU ESP32, Nesnelerin İnterneti (IoT) projeleri için ideal bir platform sağlar. Kart, düşük güç tüketimi, yüksek işlem gücü ve kapsamlı bağlantı seçenekleri gibi özelliklere sahiptir. Bu fonksiyonlar akıllı ev sistemleri, sensör ağları, uzaktan kumanda cihazları, otomasyon projeleri gibi birçok farklı uygulamada kullanılabilmektedir. ESP32, Arduino IDE gibi popüler geliştirme ortamlarıyla programlanabilir. Arduino tabanlı projelerde kullanılan kitaplıkların ve kaynakların birçoğu NodeMCU ESP32'ye de uygulanabilir. NodeMCU ESP32, genel amaçlı giriş/çıkış (GPIO) pinleri sayesinde harici sensörler, ekranlar, motorlar ve diğer bileşenlerle kolayca iletişim kurabilir. Wi-Fi ve Bluetooth bağlantı işlevleri sayesinde internete veya diğer cihazlara da bağlanabilir. Sonuç olarak, NodeMCU ESP32, IoT projeleri için güçlü ve çok yönlü bir geliştirme kartıdır. Projelerinizde kullanabileceğiniz kapsamlı bağlantı, yüksek işlem gücü ve kolay programlanabilir fonksiyonlara sahip bir platform sunar.

Ajax, "Asynchronous JavaScript and XML"in kısaltmasıdır. Web geliştirmede kullanılan bir tekniktir. Ajax, bir web sayfasında veri alışverişi yapmak için kullanılır ve sayfayı yeniden yüklemeye gerek kalmadan arka planda veri alışverişine izin verir.

Node.js, JavaScript tabanlı bir çalıştırma ortamıdır. Sunucu taraflı uygulamalar geliştirmek için kullanılır ve JavaScript'in tarayıcı ortamı dışında çalışmasına izin verir. Node.js, olay güdümlü ve eşzamansız bir mimariye sahiptir ve genellikle hızlı ve ölçeklenebilir web siteleri oluşturmak için kullanılır. Ajax ve Node.js farklı ancak tamamlayıcı teknolojilerdir. Ajax istemci tarafında (tarayıcı) çalışırken, Node.js sunucu tarafında çalışır. Ajax, kullanıcı etkileşimli web uygulamaları için kullanılırken Node.js, sunucu tarafı veri işleme ve yönetimi için kullanılır.

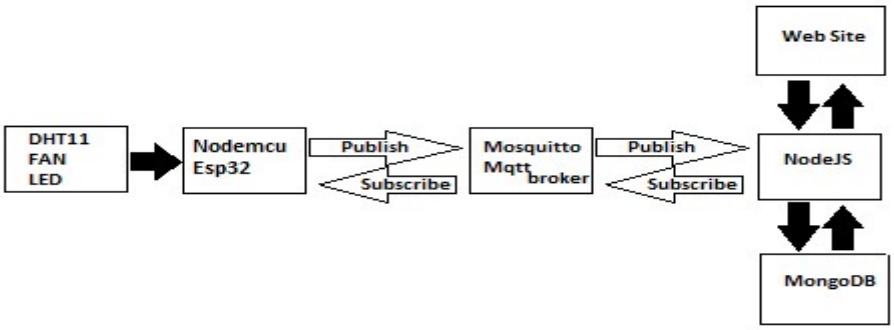
Mosquitto, açık kaynaklı bir MQTT (Message Queuing Telemetry Transport) mesaj aracısıdır. MQTT, düşük bant genişliği ve düşük güç tüketimi gerektiren cihazlar arasında

hafif ve verimli bir iletişim protokolüdür. Mosquitto, MQTT protokolünü destekleyen istemcilerin ve sunucuların birbirleriyle iletişim kurmasını sağlar. MQTT, Nesnelerin İnterneti (IoT) cihazlarının, sensörlerin ve uygulamaların birbirleriyle bilgi paylaşmasını kolaylaştırmak için kullanılır. Mosquitto, MQTT kullanarak mesaj yayınlama ve abone olma yeteneği sağlar. Bu şekilde, cihazlar ve uygulamalar veri alışverişi yapabilir ve olaylara tepki verebilir. Mosquitto, düşük kaynak tüketimi ve güçlü özellikleri ile öne çıkıyor ve birçok platformda kullanılabiliyor. Mosquitto, MQTT protokolünü uygulamak için kullanılan bir MQTT aracısıdır ve genellikle IoT projelerinde, sensör ağlarında, veri toplama sistemlerinde ve dağıtılmış iletişim senaryolarında kullanılır. Ayrıca Mosquitto, açık kaynak kodu sayesinde geniş bir geliştirici topluluğu tarafından desteklenmekte ve farklı platformlarda kullanılabilecek farklı uygulamalar geliştirmek için kullanılabilmektedir.

API, "Application Programming Interface" kısaltmasıdır ve yazılım geliştiricilerin farklı yazılım sistemlerini birbirine bağlamasını veya bir yazılım sistemi içindeki işlevleri kullanmasını sağlayan bir arayüzdür. API'ler, önceden belirlenmiş bir set işlevi içerir ve bu işlevler, bir programlama dili ile çağrılabilir. API'lerin kullanım alanı oldukça geniş olup, web tabanlı uygulamalarda, mobil uygulamalarda, otomasyon ve veri işleme sistemlerinde kullanılabilirler. API'ler, yazılım geliştirme sürecini hızlandırmak, kod tekrarını önlemek ve veri paylaşımını kolaylaştırmak için önemli bir araçtır.

Projenin amacı; Projenin amacı; donanım ve database arasında MQTT protokolünü kullanarak MongoDB database'ine verileri göndermek, gönderilen verileri API kullanarak web sitesine basmak ve web sitesinden gelen komutları donanımda gerçekleştirmektir.

Proje işleyişi Şekil 1.1'de gösterilmektedir.



Şekil 1.1: Proje İşleyisi

Proje 4 tasktan oluşmakta ve tasklar aşağıda sıralanmaktadır:

Task-1 : Sıcaklık değerlerini anlık görüntüleme

Task-2 : Nem değerlerini anlık görüntüleme

Task-3 : Open/close butonları ile led ve fan kontrolü

Task-4 : Son 5 sıcaklık değerini anlık görüntüleme

1.1 Literatür Taraması

Bu IoT projesi, Nodemcu ESP8266 WiFi modülü, MongoDB veritabanı, Mosquito, bir API ve bir web sitesi kullanılarak gerçekleştirilebilir. Bu bileşenler, cihazdan veri alınmasına, veritabanında depolanmasına, verilerin API tarafından sunulmasına ve son olarak web sitesinde gösterilmesine olanak tanır.

Benzer IoT projeleri literatürde mevcuttur. Örneğin, bir sıcaklık ve nem ölçüm sistemi, bir akıllı ev sistemi veya bir trafik ışığı kontrol sistemi, benzer bileşenlerin kullanımını içerebilir. Bu projelerin çoğu, donanım bileşenlerinin seçimi, yazılım kodlama, veri yönetimi ve sunum gibi konularda ayrıntılı bir açıklama sunar.

Bunun yanı sıra, Node.js ve Express.js gibi web uygulama geliştirme çerçeveleri kullanarak API ve web sitesi oluşturma konusunda birçok kaynak mevcuttur. MongoDB ve ESP8266 hakkında ayrıntılı bilgi ve örnekler de mevcuttur.

Bu proje için birkaç örnek aşağıdaki gibidir:

1. "IoT-based Smart Irrigation System using ESP8266, MongoDB, and Node.js" - bu proje, bir IoT tabanlı akıllı sulama sistemi oluşturmak için ESP8266, MongoDB ve Node.js kullanımını içerir. Bu proje, bitkilerin optimum nem seviyelerini koruyan bir sulama sistemi tasarlamayı amaçlamaktadır.

2. "ESP8266 and MongoDB-based Smart Home Automation System" - bu proje, bir IoT tabanlı akıllı ev otomasyon sistemi için ESP8266 ve MongoDB kullanımını içerir. Bu proje, bir evin farklı bileşenlerini (ışıklar, fanlar, vb.) uzaktan kontrol etmek için bir IoT sistemi tasarlamayı amaçlamaktadır. ESP8266 WiFi modülü ve MongoDB veritabanı kullanılarak bir IoT altyapısı oluşturulur.

3. "Traffic Signal Control System using IoT" - bu proje, bir IoT tabanlı trafik ışığı kontrol sistemi için ESP8266, MongoDB ve Node.js kullanımını içerir. Bu proje, trafik sinyallerinin IoT teknolojisi kullanılarak yönetilmesini amaçlamaktadır. ESP8266 WiFi modülü, bir trafik ışığı kontrol paneli ve bir web arayüzü kullanarak bir IoT altyapısı oluşturulur.

Bu kaynaklar, MSP430 mikrodenetleyicisi, ESP8266 WiFi modülü, MongoDB veritabanı, API ve web sitesi kullanarak bir IoT projesi oluşturmak isteyenler için faydalı bir literatür taraması sağlayabilir.

2. KULLANILAN PLATFORMLAR

2.1 Kullanılan Bilgisayar Platformları

2.1.1 Visual Studio Code

Visual Studio Code (VS Code), Microsoft tarafından geliştirilen bir kod editörüdür ve Node.js uygulamaları geliştirmek için sıklıkla kullanılır.

2.1.2 NodeJS

Node.js, JavaScript programlama dili için bir çalışma zamanı ortamıdır ve özellikle sunucu tarafı uygulamaları geliştirmek için kullanılır.

2.1.3 MongoDB Atlas

MongoDB Atlas, MongoDB veritabanı hizmetinin bulut tabanlı bir sürümüdür. Bu hizmet sayesinde kullanıcılar, MongoDB veritabanlarını bulutta yönetebilirler. Atlas, yüksek ölçeklenebilirlik, güvenlik ve kullanım kolaylığı sağlar. Ayrıca, kullanıcıların iş yüklerini kolayca yönetmelerine olanak tanıyan bir dizi özellik sunar.

2.2 Kullanılan Donanım Platformları

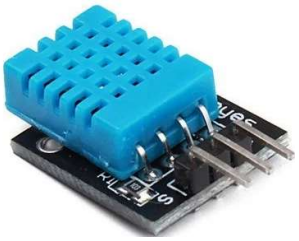
2.2.1 Nodemcu ESP32

NodeMCU ESP32, IoT projeleri için güçlü ve çok yönlü bir geliştirme kartıdır. Geniş bağlantı seçenekleri, yüksek işlem gücü ve kolay programlanabilirlik özellikleriyle projelerinizde kullanabileceğiniz bir platform sunar. Nodemcu ESP32 Şekil 2.1'de gösterilmektedir.

Şekil 2.1: Nodemcu ESP32

2.2.2 DHT11 Isı ve Nem Sensörü Kart

DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimidir. DHT11 ısı ve nem sensörü Şekil 2.2'de gösterilmektedir.



Şekil 2.2: DHT11 Isı ve Nem Sensörü Kart

2.2.3 Fan

Sıcaklık artışında soğutma işlemi için kullanılmaktadır. Fan Şekil 2.3'de gösterilmektedir.

Şekil 2.3:Fan

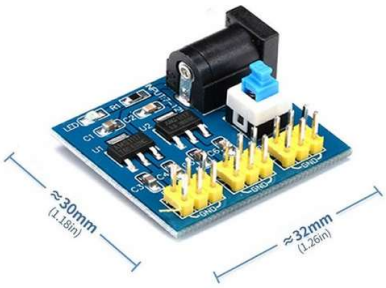
2.2.4 Led

Sıcaklık azaldığında ısıtma işlemi için kullanılmaktadır. Led Şekil 2.4'de gösterilmektedir.

Şekil 2.4:Led

2.2.5 5V 3.3V Gerilim Modülü

12V gerilimi 3,3V ve 5V gerilime düşürmek için kullanılır. 5V 3.3V Gerilim Modülü Şekil 2.5'de gösterilmektedir.



Şekil 2.5: 5V 3.3V Gerilim Modülü

2.2.6 12V Adaptör

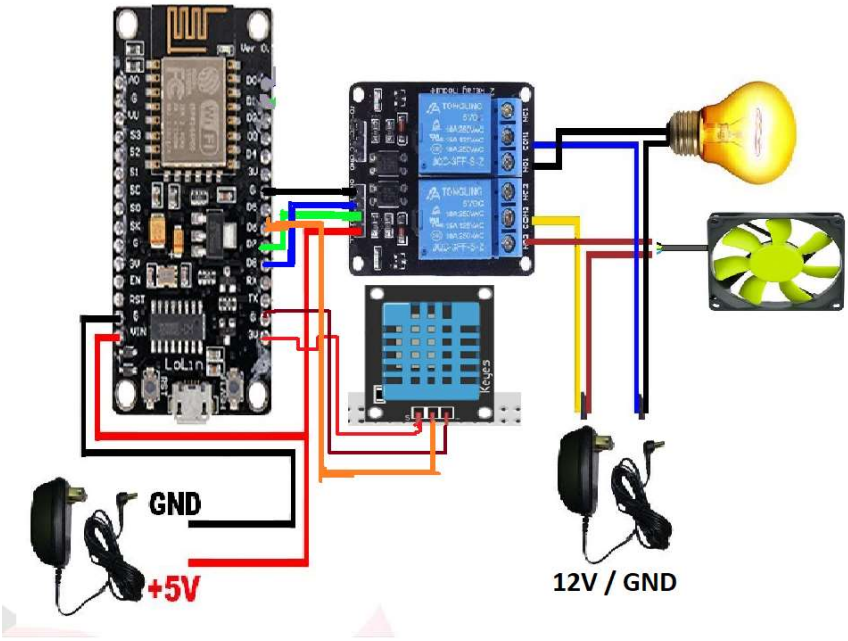
Gerekli olan enerji için kullanılır. 12V adaptör Şekil 2.6'de gösterilmektedir

Şekil 2.6: 12V Adaptör

3.PROJENİN GERÇEKLEŞTİRİLMESİ

3.1 Donanım Kısmı

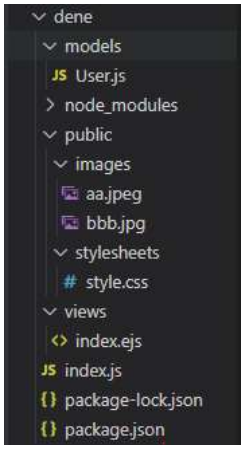
Projenin devre şematiği Şekil 3.1'de gösterilmektedir.



Şekil 3.1:Proje Devre Şeması

3.2 Yazılım Kısmı

VS Code platformunda hazırlanan kodlamaların bulunduğu klasör içeriği Şekil 3.2'de gösterilmektedir.



Şekil 3.2:Klasör içeriği

Package.json , Node.js projesinin ana yapılandırma dosyasıdır. Bu dosya, proje bağımlılıklarını, proje hakkındaki bilgileri ve proje komutlarını tanımlayan bir JavaScript Nesne Gösterimi (JSON) dosyasıdır. Package.json içeriği Şekil 3.3'de gösterilmektedir.

```json
{
  "name": "deneaa",
  "version": "1.0.0",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "build": "npm install && npm run build ",
    "start": "node index.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "cors": "^2.8.5",
    "dotenv": "^16.1.4",
    "ejs": "^3.1.9",
    "express": "^4.18.2",
    "express-ejs-layouts": "^2.5.1",
    "highcharts": "^11.0.1",
    "mongoose": "^7.2.0",
    "mqtt": "^4.3.7",
    "twilio": "^4.11.1"
  },
  "description": ""
}
```

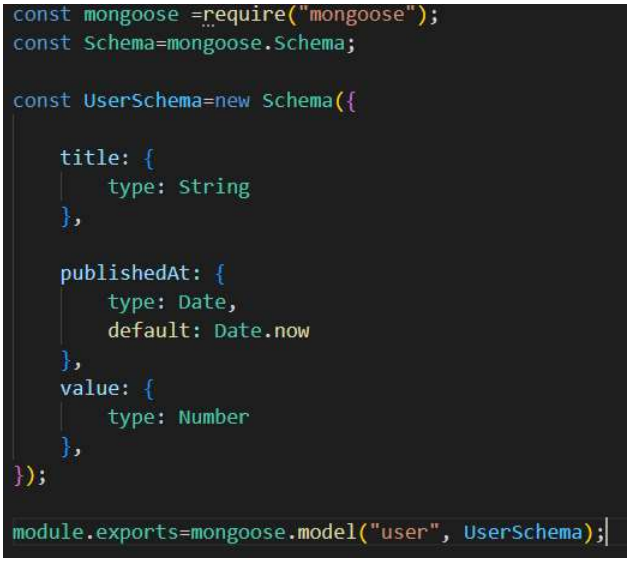
Şekil 3.3: Package.json içeriği

Node.js ile MongoDb Atlas bağlantısı sağlandı. Kod dizini Şekil 3.4'de gösterilmektedir.

```
mongoose.connect('mongodb+srv://kaplann:OVmdIOLoUqUeTN9x@cluster1.
r6o0vrc.mongodb.net/?retryWrites=true&w=majority',
{ useNewUrlParser: true,
useUnifiedTopology: true });
mongoose.connection.on('open', ()=>{
  console.log("MongoDB Connected!!!");
});
mongoose.connection.on('error',(err)=>{
  console.log("MongoDB error *XXXXX*",err);
})
```

Şekil 3.4:MongoDB Atlas Bağlantı Kodu

Mongoose şema (schema) ve modeli tanımlandı ve bunu başka bir dosyada dışa aktarıldı. Kod dizini Şekil 3.5'de gösterilmektedir.



Şekil 3.5:Mongoose Schema ve Model

MQTT istemcisine bağlandığında belirli işlemleri gerçekleştirmek için kullanılıcak kodlama yapıldı..Kod dizini Şekil 3.6'da gösterilmektedir.

Şekil 3.6:MQTT istemcisi Publish ve Subscribe

MQTT istemcisinden gelen sıcaklık ve nem değerleri MongoDB Atlas'a kaydedilmiştir. Kod dizini Şekil 3.7'de gösterilmiştir.

```
client.on('message', function (topic, message) {
    // message is Buffer
    console.log(topic.toString()+":"+message.toString())
    let data=message.toString();
    data =JSON.parse(data);
    let values=parseInt(message.toString());
    var topic1=topic.toString();
    const user=new User({
      title: topic1,
      value: values
  });
  //client.publish('ledfan', msg);

  console.log(user);
  user.save();

})
```

Şekil 3.7:MQTT ile MongoDB veri kaydı

MongoDB Atlas'a kaydedilmiş örnek veriler Şekil 3.8'de gösterilmiştir.

Şekil 3.8:MongoDB Kaydedilen veriler

Express uygulamasında '/' yoluna yönelik bir GET isteği işlevi yapıldı. İşlev, "index" adlı bir şablon dosyasını render ederek ve bunu bir yanıt olarak gönderir.Kod dizini Şekil 3.9'da gösterilmiştir

```
app.get('/',(req,res)=> {
    res.render('index');
});
```

Şekil 3.9:Get isteği ve Render

Web sayfasında led ve fan açılıp kapanması için ilgili butonlara basıldığında MQTT ile Nodemcu'ya mesaj gönderildi. Kod dizini Şekil 3.10 ve Şekil 3.11'de gösterilmiştir.

```
app.post('/update-msg', (req, res) => {
  const { value } = req.body;

  // msg değişkenini güncelle
  if (value === 'on') {
    msg = 'on';
  } else if (value === 'off') {
    msg = 'off';
  } else if (value === 'on1') {
    msg = 'on1';
  } else if (value === 'off1') {
    msg = 'off1';
  }
  console.log(msg);
  client.publish('ledfan', msg);

  res.sendStatus(200); // Başarılı yanıt dön
});
```

Şekil 3.10:Led ve Fan Aç/Kapat bilgisi

```
<div class="card-info">
  <h1 id="fan">FAN</h1>
</div>
<button id="fanOpen" class="button button1" onclick="updateMsg('on1')"  >OPEN </button>
<button id="fanClose" class="button button2" onclick="updateMsg('off1')" >CLOSE</button>
<script src="index.js" ></script>
<script>
  function updateMsg(value) {
    // Update the msg variable in index.js
    fetch('/update-msg', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ value })
    });
  }
</script>
```

Şekil 3.11:Butona basıldığında dönecek function

MongoDB'de kayıtlı son nem bilgisi anlık olarak görüntülendi. Kod dizini Şekil 3.12 , Şekil 3.13 ve Şekil 3.14'de gösterilmiştir.

```
app.get('/kisiler', function (req, res) {
  User.find({title:"humidity"},{value:1,_id:0}).sort({ _id: -1 }).limit(1).then(data2=>{
    console.log(data2);

    humdt2=data2;
  }).catch((err)=>{
    console.log(err);
  })

  res.json(humdt2);
});
```

Şekil 3.12:İndex.js'de nem verisini JSON formatında gönderme

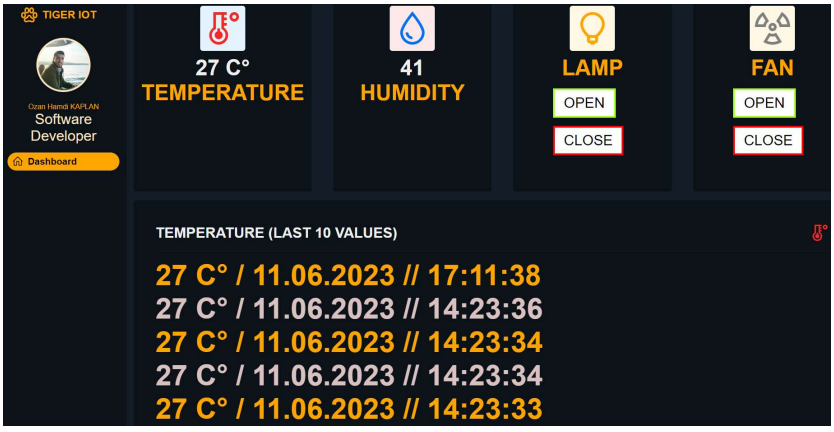
```
function getir() {
  var veriler=document.getElementById("humidity");
  ysAjax("kisiler","GET",null,function(sonuc){
    var gelenVeriler = JSON.parse(sonuc.responseText);
    veriler.innerText=gelenVeriler[0].value;
  });
}
setInterval(getir, 2000);
```

Şekil 3.13:index.ejs'de index.js'den gelen veriyi gelen Veriler değişkenine atama

```
    <h2 id="humidity" ></h2>
    <h3>HUMIDITY</h3>
  </div>
  <script>
        'use strict';
        function ysAjax(url, method, dataa, callback) {
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange=function(){
          if(this.status==200){
            if(typeof callback=='function'){
              callback(this);
            }
          }
        }
        xhr.open(method,url);
        xhr.send(dataa);
      }
```

Şekil 3.14:İndex.ejs'de ysAjax function

Projenin çalıştırıldı. Şekil 3.15'de gösterilmiştir.



Şekil 3.15:Web Görüntüleri

SONUÇ VE ÖNERİLER

Nodemcu, ESP8266, MongoDB, API ve web sitesi kullanılarak yapılan IoT projeleri oldukça geniş bir alanda uygulanabilir. Örneğin, birçok ev otomasyonu projesinde kullanılabilirler. Bu projelerde, ESP8266 kullanılarak cihazlar Wi-Fi ağına bağlanır, sensörlerden gelen veriler MongoDB'de depolanır ve API aracılığıyla web sitesi veya mobil uygulama ile erişilir.

Bu tür projelerin sonuçları oldukça başarılı olmuştur. Örneğin, "IoT-based Smart Irrigation System using ESP8266, MongoDB, and Node.js" projesinde, su tasarrufu sağlayan akıllı bir sulama sistemi oluşturulmuştur. Projenin sonuçları, su tasarrufu sağladığını ve bitki büyümesini artırdığını göstermiştir.

Benzer şekilde, "ESP8266 and MongoDB-based Smart Home Automation System" projesinde de başarılı sonuçlar elde edilmiştir. Proje, ev otomasyonunu daha akıllı hale getirerek enerji tasarrufu sağlamıştır.

Bununla birlikte, bu tür projelerin başarısı için dikkat edilmesi gereken bazı öneriler bulunmaktadır. Özellikle güvenlik konusunda önlem almak, veri bütünlüğünü korumak ve veri gizliliğini sağlamak önemlidir. Ayrıca, projenin ihtiyaçlarına uygun donanım ve yazılım seçimi de büyük önem taşır. Son olarak, projenin tasarımı ve uygulanması için iyi bir planlama yapılması ve test sürecinin yeterince uzun tutulması gerekmektedir.

KAYNAKÇA

1)https://circuitdigest.com/microcontroller-projects/iot-based-smart-irrigation-system-using-esp8266-mongodb-and-nodejs

2)https://iotdesignpro.com/projects/esp8266-and-mongodb-based-smart-home-automation-system

3)https://circuitdigest.com/microcontroller-projects/iot-based-traffic-signal-control-system

4)https://www.instructables.com/Connect-Raspberry-Pi-Pico-W-to-NodeJS-pplication-/

5)https://www.jenniferbland.com/saving-data-to-mongodb-database-from-node-js-application-tutorial/

6)https://www.mcu-turkey.com/wp-content/uploads/2011/03/msp430-programlama-notlari-uygulamalar-bilgiler.pdf

7) https://tr.wikipedia.org/wiki/Nesnelerin_interneti

8) https://www.yusufsezer.com.tr/nodejs-ajax/

9)https://wmaraci.com/forum/webmaster-genel/sayfa-yenilenmeden-verileri-guncelleme-624439.html

10) https://medium.com/@solobasay/node-js-ms-sql-ajax-f7beaa70ec7

11) https://www.w3schools.com/jsref/met_document_getelementbyid.asp