

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**MARMARİS’DE ASTIM HASTALARI İÇİN ANLIK**  
**RİSKLİ BÖLGE BİLGİLENDİRME PLATFORMU**

**Ekrem BALI**  
**Doğukan AYDOĞDU**

**Tez Danışmanı**  
**Dr.Öğr.Üyesi**  
**Güngör YILDIRIM**

**BİTİRME TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**ELAZİĞ**  
**2022**

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**MARMARİS’DE ASTIM HASTALARI İÇİN ANLIK**  
**RİSKLİ BÖLGE BİLGİLENDİRME PLATFORMU**

Ekrem BALI  
Doğukan AYDOĞDU

**BİTİRME TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Bu bitirme tezi ...../...../2022 tarihinde, aşağıda belirtilen jüri tarafından oybirliği/oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

(imza)  
Dr.Öğr.Üyesi Güngör YILDIRIM

(imza)  
Üye

(imza)  
Üye

## **ÖZGÜNLÜK BİLDİRİMİ**

Bu çalışmada, başka kaynaklardan yapılan tüm alıntılar, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini, alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

Fırat Üniversitesi

Bilgisayar Mühendisliği

Ekrem BALI  
Doğukan AYDOĞDU

## BENZERLİK BİLDİRİMİ

Gönderim Tarihi: 11-Oca-2022 02:00PM (UTC+0300)  
Gönderim Numarası: 1740043185  
Dosya adı: Dogukan-Aydogdu-Ekrem-Bal.docx (2.11M)  
Kelime sayısı: 7045  
Karakter sayısı: 46208

### Doğukan Aydoğdu Bitirme Tez

#### ORJİNALLİK RAPORU

% <b>3</b>	% <b>2</b>	% <b>0</b>	% <b>2</b>
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

#### BİRİNCİL KAYNAKLAR

1	Submitted to Fırat Üniversitesi Öğrenci Ödevi	%1
2	stemarobokod.com İnternet Kaynağı	<%1
3	Submitted to Bozok Üniversitesi Öğrenci Ödevi	<%1
4	memoria.ifrn.edu.br İnternet Kaynağı	<%1
5	Submitted to Dokuz Eylül Üniversitesi Öğrenci Ödevi	<%1
6	acikerisim.firat.edu.tr İnternet Kaynağı	<%1
7	Submitted to Istanbul Aydın University Öğrenci Ödevi	<%1
8	Submitted to Kocaeli Üniversitesi Öğrenci Ödevi	<%1
9	9lib.net İnternet Kaynağı	<%1
10	aof.sorular.net İnternet Kaynağı	<%1
11	silo.tips İnternet Kaynağı	<%1
12	www.medicine.ankara.edu.tr İnternet Kaynağı	<%1

## **TEŞEKKÜR**

Projenin tüm aşamalarında bize desteklerini, bilgi ve tecrübelerini esirgemeyen değerli hocamız Dr.Öğr.Üyesi G ng r YILDIRIM'a te ekk rlerimizi sunarız.

# İÇİNDEKİLER

## Sayfa No

<b>İÇİNDEKİLER.....</b>	<b>I</b>
<b>TABLolar LİSTESİ.....</b>	<b>II</b>
<b>ŞEKİLLER LİSTESİ.....</b>	<b>III</b>
<b>KISALTMALAR LİSTESİ .....</b>	<b>V</b>
<b>ÖZET .....</b>	<b>VI</b>
<b>ABSTRACT .....</b>	<b>VII</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KULLANILAN TEKNOLOJİLERİN TANITIMI ve KURULUMU.....</b>	<b>2</b>
2.1. Arduino Nedir ve Kurulumu .....	2
2.2. Gaz Sensörlerinin Tanıtımı .....	4
2.3. SD Kart Modülünün Tanıtımı .....	5
2.4. SIM808 GSM/GPRS/GPS Geliştirme Kartı Nedir ve Bağlantısı.....	6
2.5. Visual Studio Code Tanıtım ve Kurulumu .....	7
2.6. HTML, CSS ve JavaScript Nedir .....	8
2.7. PHP Nedir .....	10
2.8. XAMPP Nedir ve Kurulumu.....	10
<b>3. PROJENİN GERÇEKLEŞTİRME AŞAMALARI.....</b>	<b>13</b>
3.1. SIM808'den GPS Verilerinin Alınması.....	13
3.2. Arduino ile Gaz Sensörlerinin Bağlantısı .....	17
3.3. Okunan Verilerin SD Karta Yazılması .....	17
3.4. SIM808 ile SMS Gönderme.....	19
3.5. HTML ile Web Sitesi Tasarımı .....	21
3.6. CSS ile Web Sitesi Tasarımı .....	23
3.7. JavaScript ile Google Map API Kullanımı .....	29
3.8. XAMPP ile Veri Tabanının Oluşturulması.....	31
3.9. PHP ile Server Bağlantısının Kurulması .....	32
<b>4. SONUÇLAR.....</b>	<b>38</b>
<b>KAYNAKLAR.....</b>	<b>39</b>
<b>ÖZGEÇMİŞ .....</b>	<b>40</b>

## TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
<b>Tablo 2.1.</b> XAMPP Bünyesinde Bulunan Sistemler .....	11
<b>Tablo 3.1.</b> SIM808 AT Komutları .....	13
<b>Tablo 3.2.</b> SD Kart Modülü ile Arduino Mega Bağlantısı .....	18
<b>Tablo 3.3.</b> Sık Kullanılan CSS Kodları .....	24
<b>Tablo 3.4.</b> Projede Kullanılan PHPMailer Değişken ve Metodları.....	34

## ŞEKİLLER LİSTESİ

### Sayfa No

Şekil 2.1.	Arduino Mega 2560 R3 .....	2
Şekil 2.2.	Windows Store Arduino IDE İndirme Ekranı.....	3
Şekil 2.3.	Arduino Kart Seçimi .....	3
Şekil 2.4.	Arduino Port Seçme .....	4
Şekil 2.5.	MQ-135 Hava Kalite Sensör Kartı.....	5
Şekil 2.6.	MQ-07 Karbonmonoksit Gaz Sensör Kartı .....	5
Şekil 2.7.	SD Kart Modülü.....	5
Şekil 2.8.	SIM808 GSM/GPRS/GPS Geliştirme Kartı .....	6
Şekil 2.9.	SIM808 Arduino Bağlantısı.....	7
Şekil 2.10.	Visual Studio Code Kurulum Yardımcısı.....	8
Şekil 2.11.	Visual Studio Code .....	8
Şekil 2.12.	Varsayılan Bir HTML Yapısı .....	9
Şekil 2.13.	XAMPP Kurulum Sihirbazı.....	11
Şekil 2.14.	XAMPP Kontrol Paneli .....	12
Şekil 3.1.	Analog Pinlerin Tanımlanması .....	14
Şekil 3.2.	Arduino ile SIM808 Haberleşme Metodları .....	15
Şekil 3.3.	Arduino Konum Bulma Metodu .....	16
Şekil 3.4.	Arduino Konum Bulma Metodunun Devamı .....	16
Şekil 3.5.	Arduino’da GPS Verisini Bölen Metod.....	17
Şekil 3.6.	Arduino Setup Metodu .....	18
Şekil 3.7.	Arduino’da SD Karta Yazma Metodu .....	19
Şekil 3.8.	Arduino’da Gelen SMS’in Kontrolü .....	20
Şekil 3.9.	Arduino’da SMS Gönderen Metod .....	20



<b>Şekil 3.10.</b>	Kurulan HTML Yapısı ve Header Bölümü .....	21
<b>Şekil 3.11.</b>	Astım Bilgi Bölümünün HTML Yapısı.....	22
<b>Şekil 3.12.</b>	Harita Bölümünün HTML Yapısı .....	22
<b>Şekil 3.13.</b>	Abonelik Bölümünün HTML Yapısı .....	23
<b>Şekil 3.14.</b>	CSS ile HTML ve Body Etiketlerinin Tasarımı .....	24
<b>Şekil 3.15.</b>	CSS ile Header Bölümünün Tasarımı .....	25
<b>Şekil 3.16.</b>	CSS ile Astım Bilgi Bölümünün Tasarımı .....	26
<b>Şekil 3.17.</b>	CSS ile Harita Bölümünün Tasarımı.....	26
<b>Şekil 3.18.</b>	CSS ile Abonelik Formunun Tasarımı .....	27
<b>Şekil 3.19.</b>	CSS ile Buton ve Footer Tasarımı.....	28
<b>Şekil 3.20.</b>	CSS ile Responsive Tasarım.....	28
<b>Şekil 3.21.</b>	CSS ile 750 px İçin Responsive Tasarım .....	29
<b>Şekil 3.22.</b>	Haritaya Çizilen Veriler .....	30
<b>Şekil 3.23.</b>	Verilerin Haritaya Çizilmesi .....	31
<b>Şekil 3.24.</b>	Aboneler Tablosu .....	32
<b>Şekil 3.25.</b>	Konum ve Sensör Verilerinin Tutulduğu Tablo.....	32
<b>Şekil 3.26.</b>	Veri Tabanı Bağlantısı.....	33
<b>Şekil 3.27.</b>	PHPMailer Kütüphanesinin Eklenmesi ve Mail Ayarları.....	34
<b>Şekil 3.28.</b>	SQL Komutları ve Değişkenler.....	35
<b>Şekil 3.29.</b>	Konum ve Sensör Verilerinin Alınması .....	36
<b>Şekil 3.30.</b>	Mail İçeriğinin Oluşturulması .....	37
<b>Şekil 3.31.</b>	Mailin Gönderilmesi .....	37

## **KISALTMALAR LİSTESİ**

<b>SMS</b>	: Short Message Service
<b>GPS</b>	: Global Positioning System
<b>GSM</b>	: Global System for Mobil
<b>SD</b>	: Secure Digital
<b>PWM</b>	: Pulse Width Modulation
<b>SPI</b>	: Serial Peripheral Interface
<b>API</b>	: Application Programming Interface
<b>HTML</b>	: Hypertext Markup Language
<b>W3C</b>	: World Wide Web Consortium
<b>CSS</b>	: Cascading Style Sheets
<b>PHP</b>	: Hypertext Preprocessor
<b>RMC</b>	: Recommended Minimum Specific
<b>SMTP</b>	: Simple Mail Transfer Protocol

## ÖZET

Günümüzde teknolojinin ve internetin gelişmesi birçok sektör için teknolojinin kullanımına olanak sağlamıştır. Bu teknolojilerden en çok faydalananlardan biri sağlık sektörüdür. Yoğun hayatlarımızda sağlık durumumuza yeterince dikkat edemeyebiliyoruz ve bu durumlarda sağlık alanlarında yapılan teknolojiler bizler için biçilmiş kaftan. Günümüzde hava kirliliğinin ve zararlı gazların artmasıyla aynı zamanda gelişen teknolojiyle bu gazları sensörler aracılığıyla tespit ve takip etmemiz çok daha kolaylaştı. Yapılan bu proje anlık riskli bölge bilgilendirme platformudur. Bu projede astım hastalarının olumsuz etkilendiği gazları hedef alarak, belirli bölgelerden gaz sensörleri yardımıyla hava kirliliği verilerini kayıt edip harita üzerinden işlemek ve kullanıcılara bu bölgelerin risk bildirimini yapmak amaçlanmıştır. Projenin genel tasarımı düşünüldü ve gereksinimleri belirlendi. Bunlar yazılım ve donanım olarak ikiye ayrılıyor. Donanımsal ihtiyaçlar: Arduino Mega, SIM808 Geliştirme Kartı, MQ-135 Hava Kalite Sensörü, MQ-07 Karbonmonoksit Gazı Sensörü. Mikro SD Kart modülü. Yazılımsal olarak gereksinimler ise kullanılan elektronik donanımları kullanabilmek için açık kaynaklı geliştirme platformu olan Arduino IDE'dir. Bunları web sitesinde görüntüleyebilmek için ise Visual Studio Code programı kullanıldı. Öncelikle Arduino Mega ile gaz sensörleri olan MQ-135 ve MQ-07 bağlantısı yapıldı ve Arduino IDE programında gerekli bağlantı kodları yazıldı. Mikro SD Kart modülü ile birkaç bölgeden aldığımız verileri saklamak için gerekli bağlantılar yapıldı ve Arduino ortamında yazıldı. Sonrasında kullanıcının istediği zaman telefonundan SMS yoluyla konum bilgisini alabilmesi için kullanılan SIM808'in Arduino ile bağlantısı yapıldı ve aynı zamanda haberleşmek için gerekli kodlar Arduino ortamında yazıldı. Kullanıcılar abone olduğunda kendilerine mail yoluyla riskli bölgelerin bildirilmesi ve dilediklerinde web sitesine girip anlık riskli bölgeleri görebilmeleri için Visual Studio Code'da gerekli kurulumlar yapıldıktan sonra bir web site tasarlandı. Abone olan kullanıcı mail yoluyla ve dilerse telefondan SMS yoluyla riskli bölgenin konumunu alabilecek.

**Anahtar Kelimeler :** Astım, Hava Kirliliği, Gaz Sensörleri, Arduino, Visual Studio Code

## ABSTRACT

Nowadays, the development of the technology and the internet enables the use of technology for many different sectors. One of the sectors which takes advantage of the technology at most is health sector. We are not be able to pay attention to our health in our busy lives. Therefore, technologies that are improved for the healthcare field are the very things for us. Nowadays, not only the increase in air pollution and harmful gases but also the developing technology make it easier to determine and follow these gases through detecting elements. This developed project is an instantaneous informing platform for risk areas. It is aimed in this project that targeting gases through which asthma patients are affected negatively, recording air pollution data with the help of gas sensors set in specific areas and processing them over a map, and notifying users for these risk areas. It is thought about the project's general design and the requirements are specified. These requirements are divided into two groups as software and hardware. Hardware requirements are Arduinio Mega, SIM808 Development Board, MQ-135 Air Quality Sensor, MQ-07 Carbon Monoxide Gas Sensor, and Micro SD Card Module. The only software requirement is Arduino IDE which is an open-sourced development platform enabling us to use electronic equipments. Visual Studio Code program is used to view these in the website. Firstly, it is made a connection with Arduino Mega and MQ-135 and MQ-07, which are gas sensors. Then, required connection codes are coded in Arduino IDE program. In order to store data gotten from some areas, required connections are made with Micro SD Card Module and coded in Arduino. After then SIM808, which is used for enabling users to get location information whenever they want from their phones through SMS, is connected with Arduino. At the same time, required codes in order to communicate are coded in Arduino. A website is designed after required installations are made in Visual Studio Code with the aims of informing risk areas through an e-mail and enabling users to enter the website and see the instantaneous risk areas whenever they want when they subscribed. Subscribed users can get the location of risk areas through an e-mail, or through an SMS if they want.

**Key Words:** Asthma, Air Pollution, Gas Sensors, Arduino, Visual Studio Code.

## 1. GİRİŞ

Günümüz dünyasında ki teknolojik gelişmeler şehir hayatını ve endüstrileşmeyi son yüzyıl içerisinde hızla arttırmıştır. Bu artış ise aynı oranda hava kirliliğinin artmasına sebep olmuştur. Hava kirliliğinin etkilediği organların başında akciğerler gelir. Solunan havada bulunan zararlı gazlar ve partiküller akciğer hastalıklarının başlamasında ve tetiklenmesinde önemli faktörlerdir.

Proje konusu olan astım hastalığı da bu hastalıklar arasındadır. Astım hastaları da hava kirliliğine sebep olan bu gazlardan olumsuz etkilenir. Azotdioksit(NO<sub>2</sub>), sülfürdioksit(SO<sub>2</sub>), karbondioksit(CO<sub>2</sub>), karbonmonoksit(CO), ozon gibi gazlar bunlara örnektir.

Bu bilgiler ışığında var olan hava kirliliğine çözüm sunulmasa da bir önlem olarak bu platform geliştirilmiştir. Astım hastalarının etkilendiği gazlar üzerinde genel bir hava kirliliği ölçümü alan MQ-135 ve havadaki karbonmonoksit miktarını ölçen MQ-07 gaz sensörleri kullanılmıştır. Bu sensörler üzerinden analog bilgiye erişmek ve işlemek için Arduino Mega geliştirme kartı kullanılmıştır. Hava kirliliği verilerinin bulunulan bölge ile kaydedilmesi ve platform üzerinden bilgi almak isteyenlere SMS gönderilebilmesi için SIM808 GSM/GPRS/GPS geliştirme kartı kullanılmıştır. Belirli bölgelerde veriler elde edilmiş ve SD kart modülü yardımı ile kaydedilmiştir.

Platform için verilerin toplanıp işlendiği Arduino kısmının dışında bir de kullanıcıya sunmak üzere bir web sayfası tasarlanmış ve veri tabanı ile bağlantılı programlanmıştır. Bu web sayfasında SIM808'den alınan GPS verileri ve gaz sensörlerinden alınan veriler Google MAP API aracılığı ile harita üzerinde işlenmiştir. Ayrıca bir abonelik formu oluşturulmuş, bu formu dolduran kullanıcının mail adresine tehlikeli ve dikkatli olunması gereken bölgelerin mail olarak atılması sağlanmıştır.

Astım hastalığı olan veya bu tür akciğer hastalıklarından korunmak isteyen ve Marmaris'de yaşayan vatandaşlar, anlık olarak gitmek istedikleri yerlerin risk analizine bakıp riskli bölgelerden uzak durarak sağlık durumlarını kontrol altında tutacaktır. Bu şekilde hava kirliliğine çözüm olmasa da astım ve hava kirliliğinden olumsuz etkilenen diğer akciğer hastalıklarının ilerleyişinin önüne geçmek amaçlanmıştır.

## 2. KULLANILAN TEKNOLOJİLERİN TANITIMI ve KURULUMU

### 2.1. Arduino Nedir ve Kurulumu

Arduino, 2003 yılında Massimo Banzi ve Casey Reas isimli iki arkadaşın öncülüğünde İtalya'nın Ivrea kasabasında bulunan Ivrea Etkileşim Tasarımı Enstitüsü'nde beş arkadaş tarafından bir üniversite projesi olarak geliştirildi.

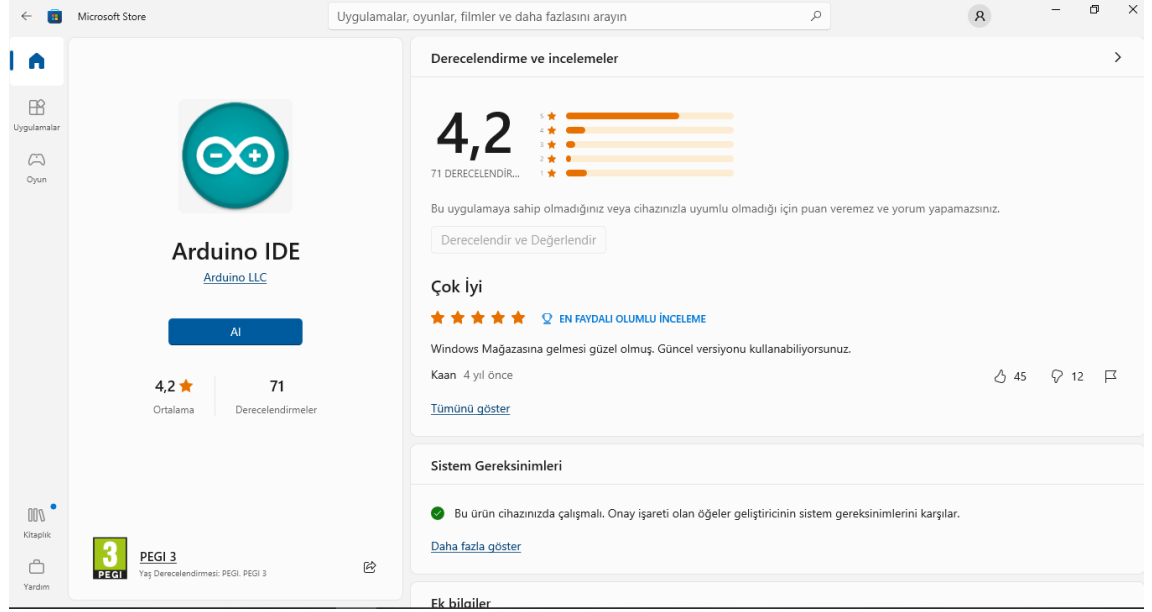
Arduino elektronik projeler oluşturmak için açık kaynaklı mikrokontrolcü platformudur. Açık kaynaklı olmasının üzerinde durmuşlardır ve bunu felsefe haline getirmişlerdir çünkü istedikleri herkesin birbirlerinden yararlanmasıdır. Arduino'nun başlıca yaratıcılarından olan Massimo Banzi, “Bizim çok iyi bir fikrimiz vardı ve bunu herkese hediye ederek zenginleştik” dediğinden de anlaşıldığı üzere her şeyin açık kaynak olmasının üzerinde durmuşlardır. Arduino'nun ihtiyaç ve gereksinimlere göre ayrılmış çeşitleri vardır. Bunlardan bazıları: Arduino Uno, Arduino Nano, Arduino Mega'dır. Arduino ile büyük veya küçük birçok proje yapılabilir. Projede kullandığımız Arduino modeli Arduino Mega 2560 R3 olduğu için teknik donanım detaylarını bu kart üzerinden görelim. Arduino Mega'nın en son çıkan modelidir. Çalışma Gerilimi 5V'dur. Şekil 2.1'de gördüğümüz üzere 54 tane Dijital giriş/çıkış pini vardır, bunlardan 15 tanesi PWM çıkışı içindir. 16 tane Analog giriş pini vardır. Her giriş ve çıkış için akım 40 mA'dir.

Arduino'nun en önemli özelliklerinden biri her kitleye hitap etmesidir. Arduino kullanmanın başlıca avantajları; programlanması kolay, kolay ulaşılabilir olması, düşük maliyet, açık kaynak sayesinde hızlı proje geliştirebilme bunlardan bazılarıdır.



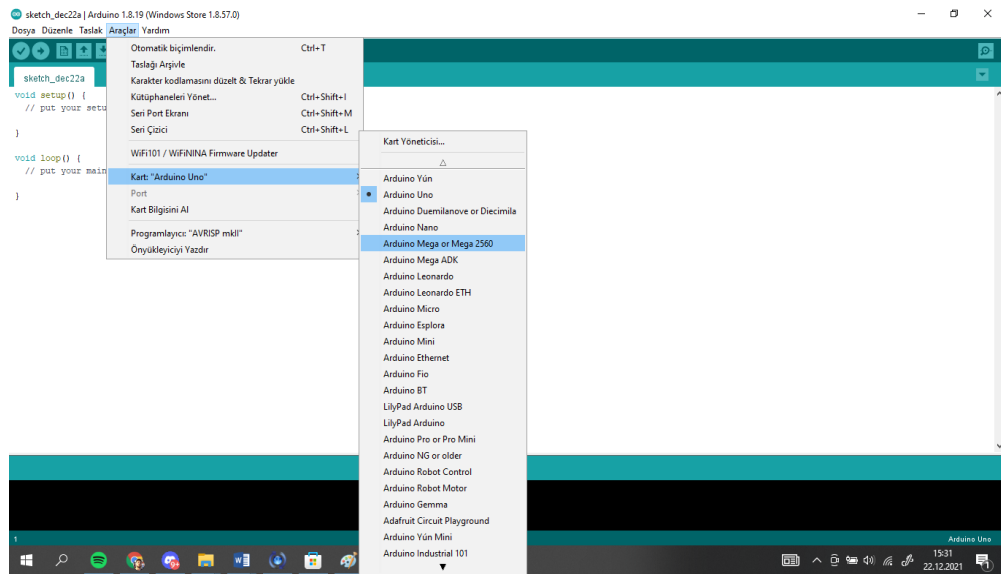
Şekil 2.1. Arduino Mega 2560 R3

Arduino IDE'yi dilerseñiz bilgisayarınız hangi işletim sistemine sahipse resmi sitesinden indirebilirsiniz. Dilerseñiz de Windows işletim sistemini kullanıyorsanız Windows Store'dan ücretsiz bir şekilde indirebilir ve kurulumla uğraşmadan direkt ara yüze ulaşabilirsiniz.



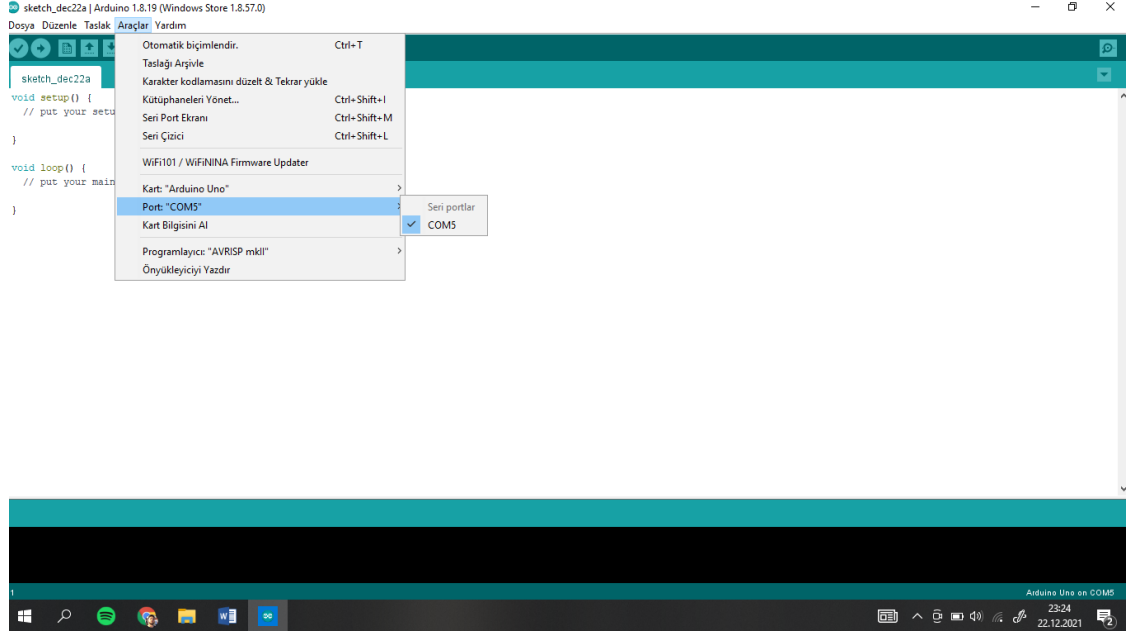
Şekil 2.2. Windows Store Arduino IDE İndirme Ekranı

Kurulumu başarıyla tamamladıktan sonra programı çalıştırdığımızda ara yüz ekranı gelecektir. Öncelikle bu bölümde sahip olduğumuz kartı programa tanıtmamız gerekiyor. Üst kısımda bulunan araçlar sekmesine basıp kullandığımız kartı seçiyoruz.



Şekil 2.3. Arduino Kart Seçimi

Arduino IDE programını açtığımızda port kontrolü yapmamız gerekiyor. Araçlar bölümünden portumuzu seçmemiz gerekiyor. Portun seçili olduğundan emin olduktan sonra Arduino'da yükle seçeneğine basarak kodumuzu Arduino kartına yüklüyoruz eğer hata gelmediyse Arduino kurulumu tamamlanmış demektir.



Şekil 2.4. Arduino Port Seçme

## 2.2. Gaz Sensörlerinin Tanıtımı

Piyasada hali hazırda birçok gaz sensörü bulunmaktadır. Bu sensörler gereksinimlere göre kullanılmaktadır. Genel olarak çalışma yapıları benzerdir. Gazı algılamak için hassas bir tel, ısıtıcı eleman ve bir adet yük direnci bulunmaktadır. Isıtıcının verdiği ısıyla ısınan metan gazı, sensörde bulunan telden geçerek telin direncinin değişmesi ile bize bilgi aktarır. Bu projede 2 farklı gaz sensörü kullanıldı bunlar: MQ-135 ve MQ-07'dir.

MQ-135 hava kalite sensör kartı NH<sub>3</sub>, NO<sub>x</sub>, benzen, alkol buharı, duman ve CO<sub>2</sub> gazlarını tespit edip bu gazların ortalamasının verisini veren bir ölçüm modülüdür. Şekil 2.4'de görüldüğü gibi sensör taşıyıcı kartı ile birlikte. Çalışma gerilimi 5V'dur. Dijital ve analog çıkışlara sahiptir.





**Şekil 2.5.** MQ-135 Hava Kalite Sensör Kartı

MQ-07 Karbonmonoksit gaz sensör kartı 10ppm ve 10.000ppm arasında Karbonmonoksit algılar. Bulunan ortamdaki gaz yoğunluğuna göre analog voltaj çıkışı verir. Çıkış gerilimi 5V'dur. 300ppm ve 10.000ppm arası tehlikeli aralık, gaz kaçağı vardır denilebilir.



**Şekil 2.6.** MQ-07 Karbonmonoksit Gaz Sensörü Kartı

### 2.3. SD Kart Modülünün Tanıtımı

SD kart modülü SPI protokolünü kullanarak, kolay bir şekilde SD kartlara okuma yazma yapabileceğiniz bir modüldür. Verileri saklama, gelen bilgileri kayıt etme, okunan bilgileri SD karta yükleme gibi birçok şekilde kullanılabilir. Kullanılan SD kart modülünde 3.3V regülatörler bulunmaktadır. 5V'luk sistemlerin Haricinde 3.3V'luk sistemlerde de rahat bir şekilde kullanılabilir.

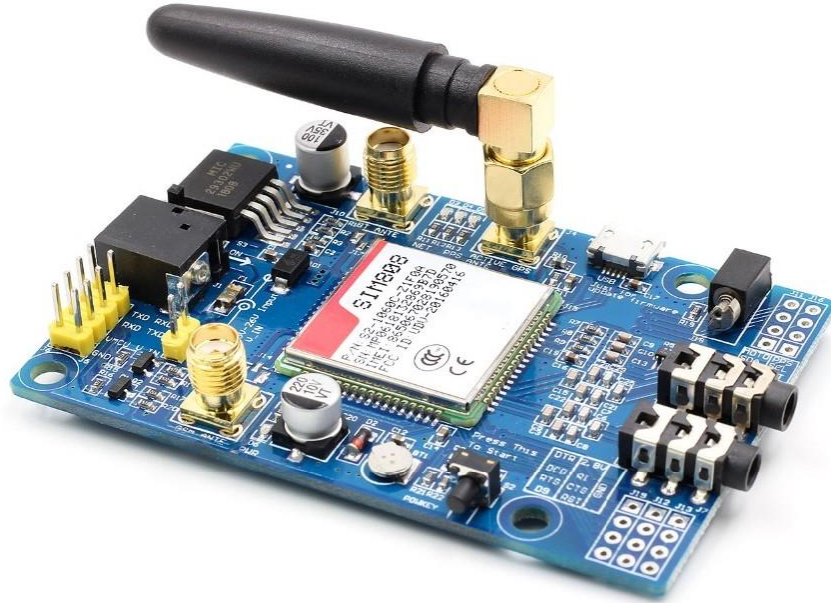


**Şekil 2.7.** SD Kart Modülü

#### 2.4. SIM808 GSM/GPRS/GPS Geliştirme Kartı Nedir ve Bağlantısı

SIM808, Arduino ve benzeri mikrodenetleyicileri GSM/GPRS/GPS özellikleriyle kullanmanızı sağlayan bir geliştirme kartıdır. Kart üzerinde ki GPS anteni sayesinde konum takibi yapılabilir. GSM özelliği ile de AT komutlarını kullanarak bir numarayı arayabilir ve mesaj atıp mesaj alabilir. Kartın üzerinde ki power düğmesi modülü açıp kapatmaya yarar. Power ON durumunda ise Power Led yanar. Modülü aktif hale getirmek için de yan tarafındaki butona 2 saniye basmak yeterli olacaktır. GSM, GPS antenleri ve SIM kartı doğru takıldıktan sonra led yanıp sönecektir. Bu da SIM808 modülünün şebekeye sorunsuz bağlandığını gösterir.

Karta üç farklı şekilde besleme yapılabilir. Bunlardan biri 5-26VDC güç kaynağı, biri VIN girişinden 5-26V, bir diğeri ise tek hücreli 3.7-4.2V Lipo/Lion pildir.



Şekil 2.8. SIM808 GSM/GPRS/GPS Geliştirme Kartı

SIM808 modülünü Arduino geliştirme kartı ile bağlantısı için, Seri haberleşme hızını 9600 baud olarak ayarladıktan sonra 3.seri haberleşme pinleri olan 14.pin RX ve 15.pin TX'i de 9600 baud olarak ayarlamalıyız. Şekil 2.9'da görüldüğü gibi loop metodunda seri iletişim kontrol edilir. İki cihaz arasında iletişim var ise Arduino'dan gelen veri SIM808'e yazılır, SIM808'den gelen de seri port ekranına yazılır. Böylece iletişim başarıyla tamamlanmış olur.

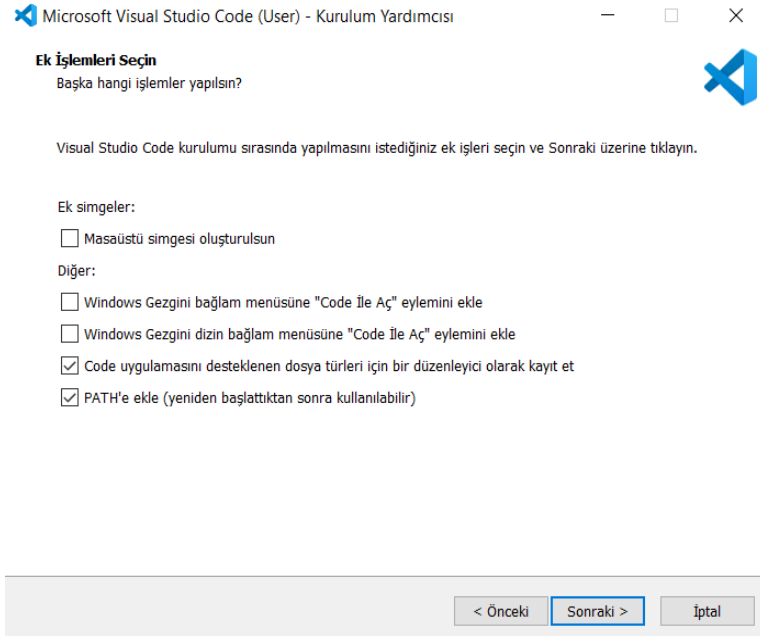


Şekil 2.9. SIM808 Arduino Bağlantısı

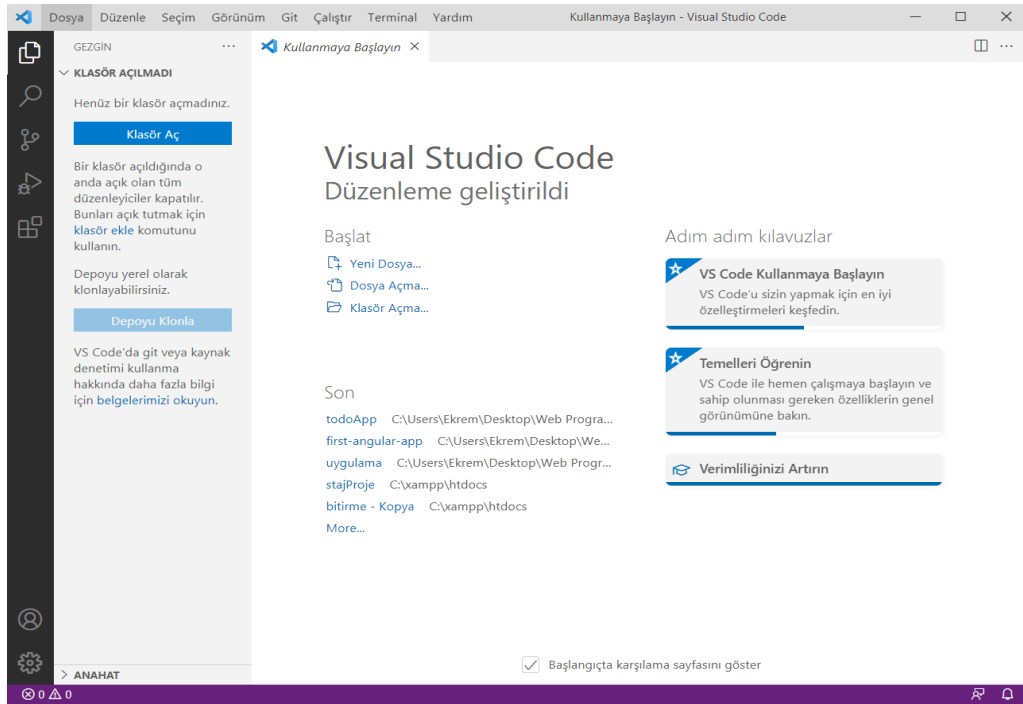
## 2.5. Visual Studio Code Tanıtım ve Kurulumu

Visual Studio Code Microsoft tarafından geliştirilen ücretsiz bir kaynak kod düzenleme editörüdür. Bünyesinde bulunan uzantılar desteği sayesinde, farklı programcılar veya şirketler tarafından geliştirilen destek uzantıları ile birçok metin işaretleme ve programlama dili için hata ayıklama, sözdizimi vurgulama, otomatik düzeltme desteği sunmaktadır. Özellikle hızlı ve hafif olması için geliştirilen editör, programcıya platformdan bağımsız, son derece basit ve hızlı bir kodlama deneyimi sunar. Birçok yazılım desteğini uzantılar üzerinden sağladığından gereksiz dosyaların indirilmesinin önüne geçer, böylece sadece ihtiyaç duyulduğunda uzantıların edinildiği hafif ve gereksiz yer tutmayan bir editör olmayı başarmıştır.

Visual Studio Code'u resmi internet sitesinden işlemcimizle uyumlu bit desteğini ve uygun işletim sistemini seçip indiriyoruz. Daha sonra indirdiğimiz kurulum sihirbazını açıp ilk sayfada sözleşmeyi kabul ediyoruz. Daha sonra program yolunu seçiyoruz. Ardından karşımıza şekil 2.10 çıkıyor buradan gerekli bulduğumuz seçenekleri seçip kuruyoruz. Bu adımdan sonra kurulum tamamlanıyor ve artık Visual Studio Code' u açıyoruz. Şekil 2.11'de gördüğümüz ana ekran karşımıza çıkıyor böylece editörü kullanmaya başlayabiliriz.



Şekil 2.10. Visual Studio Code Kurulum Yardımcısı



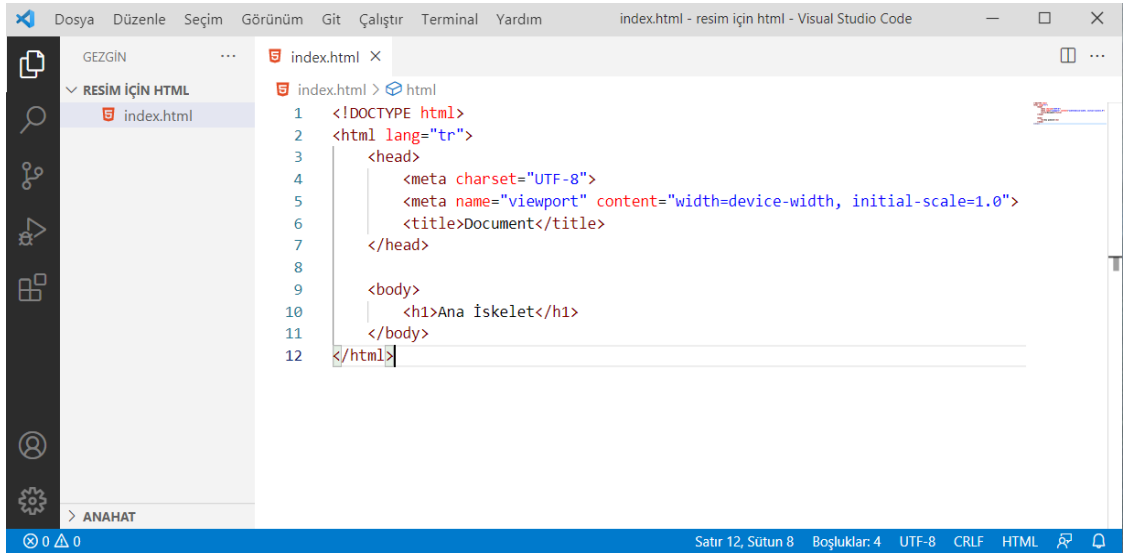
Şekil 2.11. Visual Studio Code

## 2.6. HTML, CSS ve JavaScript Nedir

HTML (Hypertext Markup Language) yani Hiper Metin İşaretleme Dili internet sayfası tasarlamada standart haline gelmiş bir işaretleme dilidir. HTML programlama dili değildir,

HTML ile kendi başına çalışan programlar yazılamaz. HTML dilinin anlaşılması için bir yorumlayıcı yani tarayıcı gerekmektedir. Bu işaretleme dili ile web sayfaları oluşturulmasının yanı sıra arama motorlarına bu sayfalar hakkında bilgiler verilmesi de sağlanır. Sayfa üzerinde yazı, resim ve video gibi verilerin görüntülenmesini ve istenilen şekilde konumlandırılmasını sağlar. Yani HTML web sayfalarının temelidir.

HTML 1990 yılında Tim Berners-Lee tarafından geliştirilmiştir. Tim Berners-Lee'nin aynı zamanda World Wide Web (WWW) sistemini kurması ile HTML 1990'lardan günümüze kullanılmaya devam edilmiş ve web sayfalarının iskeleti olmuştur. W3C tarafından geliştirilmeye devam eden HTML en son sürümü HTML5, 2014 yılında tutarlı bir W3C tavsiyesi olarak yayınlanmıştır. Şekil 2.12'de varsayılan bir HTML yapısı görülmektedir.



```
1 <!DOCTYPE html>
2 <html lang="tr">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7   </head>
8
9   <body>
10    <h1>Ana İskelet</h1>
11  </body>
12</html>
```

**Şekil 2.12.** Varsayılan Bir HTML Yapısı

CSS (Cascading Style Sheets) yani Basamaklı Stil Şablonları, HTML'e yardımcı olmak üzere, metin ve format biçimlendirme, tasarımsal renk, büyüklük vb. unsurların daha detaylı olarak düzenlenmesini sağlamak için geliştirilmiş bir işaretleme dilidir. Bunun yanı sıra metinlerin ve blokların konumlandırılmasını da sağlar. CSS programcıya büyük esneklik sağlar, herhangi bir sınıf, id veya etiket üzerinde istediğiniz biçimlendirme işlemini yapabilirsiniz. Böylece her bir metne ayrı ayrı ulaşmak zorunda olmadan bir sınıf veya etiket üzerinden ulaşabilirsiniz.

CSS'in geliştirilmesi W3C yani Worl Wide Web Consortium bünyesinde bulunan CSS çalışma grubu tarafından yapılmaktadır. Bu grup tarafından gerçekleştirilen teknik geliştirmeler W3C üyeleri tarafından değerlendirilir ve onaylandıktan sonra uygulanır. CSS ilk olarak 1996

yılında kullanıma sunulmuştur. CSS2 1998 yılında ve günümüzde de kullanılmaya devam edilen CSS3 1999 yılında kullanıma sunulmuştur.

JavaScript, tarayıcı tarafında çalışan bir diğer dildir. HTML ve CSS'in aksine dinamik, nesne yönelimli ve fonksiyonel bir betik dilidir. Web sayfası içerisinde kullanıcının etkileşime girmesini gerektiren bölümlerin programlanması için kullanılır. HTML sınıf, id ve etiketlerine ulaşılabilirdiğinden, gerek HTML etiketlerinde gerekse CSS içeriklerinde değişiklikler yapılabilmesini sağlar. Aynı zamanda http isteği oluşturabildiğinden web üzerinden veri çekip işleyebilir. Asenkron bir şekilde server ile iletişime geçebilmektedir.

JavaScript, markalaşmış bir betik dili spesifikasyonu olan ECMAScript belirtileri ile standart haline getirilmiştir. Brendan Eich tarafından geliştirilmiş ve ilk defa 1995 yılında piyasaya sürülmüştür. 1996'da JavaScript'in endüstri standardı olması için Netscape firması tarafından Ecma International'a başvurulmuş, böylece ECMAScript olarak standardize edilmiştir. JavaScript 1996'dan günümüze hâlâ ECMAScript standartlarıyla geliştirilmeye devam etmektedir.

## **2.7. PHP Nedir**

PHP, sunucu tarafı çalışan, geniş kullanımlı ve çok amaçlı bir betik ve programlama dilidir. HTML içerisine gömülebildiğinden, tek bir dosyada HTML ile tasarım yapılırken, aynı zamanda PHP ile sunucu tarafında işlem yapılabilir. PHP, sunucu tarafında en popüler dillerden biridir. PHP işleme modülüne sahip web sunucuları PHP kodlarını yorumlar, sonuç olarak da bir web sayfası üretilir. PHP ile grafiksel ara yüze sahip masaüstü uygulamalar da geliştirilebilmektedir. Formdan veri çekmek, sunucu tarafında dosyalar üzerinde değişiklikler yapmak, veri tabanlarını yönetmek ve veri tabanlarında sorgu çalıştırıp yanıtlarını almak PHP'nin temel işlevleri arasındadır.

PHP, Rasmus Lerdorf tarafından 1994 yılında, kişisel web sitesini geliştirmek için Perl betikleri yazması ile geliştirilmeye başlanmıştır. Rasmus Lerdorf sitesinin ziyaretçi trafiğini kaydetmek için bu işe başlamıştır. İlk olarak PHP/FI adı verilmiş ve 1997'de PHP/FI 2 çıkmıştır. Zeev Suraski ve Andi Gutmans tarafından geliştirilmeye devam edilmiş ve 1998'de PHP 3 duyurulmuştur. O günden beridir PHP geliştirilmeye devam etmektedir.

## **2.8. XAMPP Nedir ve Kurulumu**

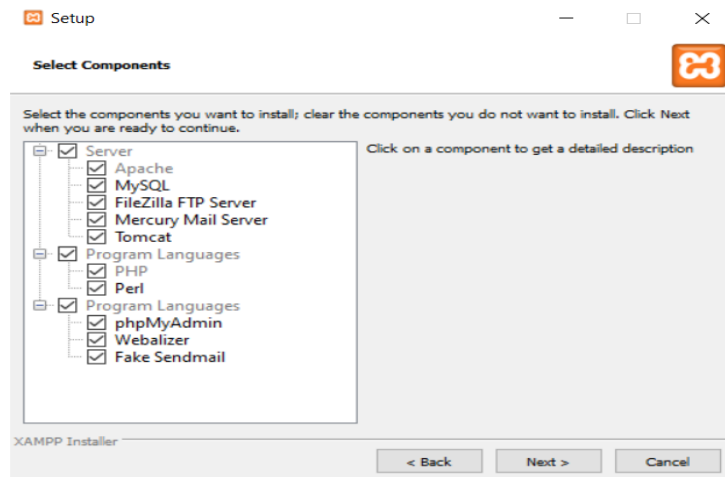
XAMPP, gerekli bileşenleri bünyesinde toplayan, son derece yararlı, web sunucusu kurmaya yarayan bir yazılım sistemidir. PHP ile çalışan XAMPP, web hizmetleri için güzel bir geliştirme ortamı sağlamaktadır. Web sunucularının büyük bir çoğunluğu benzer sistemleri

kullandığından, yerel sunucudan canlı sunucuya geçiş kolaylıkla sağlanmaktadır. XAMPP ile kurulan bir web sunucu dışarıdan erişilemez olur yalnızca kurulu sunucu üzerinde çalışır. Apache, MySQL, PHP ve Perl sistemlerini bünyesinde bulundurur. Bu sistemlerin işlevleri tablo 2.1’de verilmiştir.

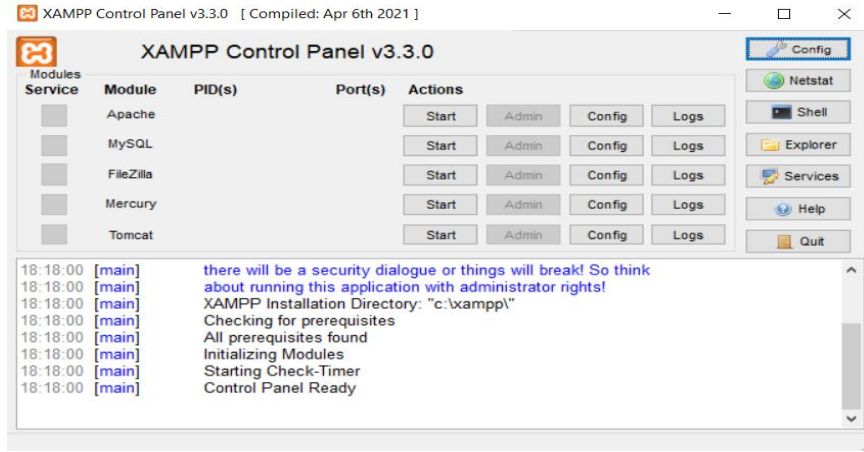
**Tablo 2.1.** XAMPP bünyesinde bulunan sistemler

Apache	Açık kaynak kodlu web sunucusu programıdır. Görevi http isteklerini işlemektir.
MySQL	Veri tabanı sistemini yönetmek için kullanılır. Verilerin depolanması ve istenilen şekilde yönetilmesini sağlar.
PHP	PHP, sunucu taraflı çalışan, geniş kullanımlı ve çok amaçlı bir betik ve programlama dilidir.
Perl	Web geliştirme ve ağ programlama yapabilen metin düzenleme için kullanılan yüksek seviye bir programlama dilidir.

XAMPP’i kurmak için öncelikle resmi sitesine girip uygun bit desteği ve işletim sistemini seçerek uygulamayı indiriyoruz. İlk olarak şekil 2.13’de görüldüğü gibi gerekli bileşenleri seçiyoruz. Ardından programın kurulacağı dosya yolunu ve tercih edilen dili seçiyor ve kurulumu tamamlıyoruz. Program kullanılmaya hazır, şekil 2.14’de XAMPP uygulamasının ana ekranını görüyoruz.



**Şekil 2.13.** XAMPP Kurulum Sihirbazı



Şekil 2.14. XAMPP Kontrol Paneli



### 3. PROJENİN GERÇEKLEŞTİRME AŞAMALARI

#### 3.1. SIM808'den GPS Verilerinin Alınması

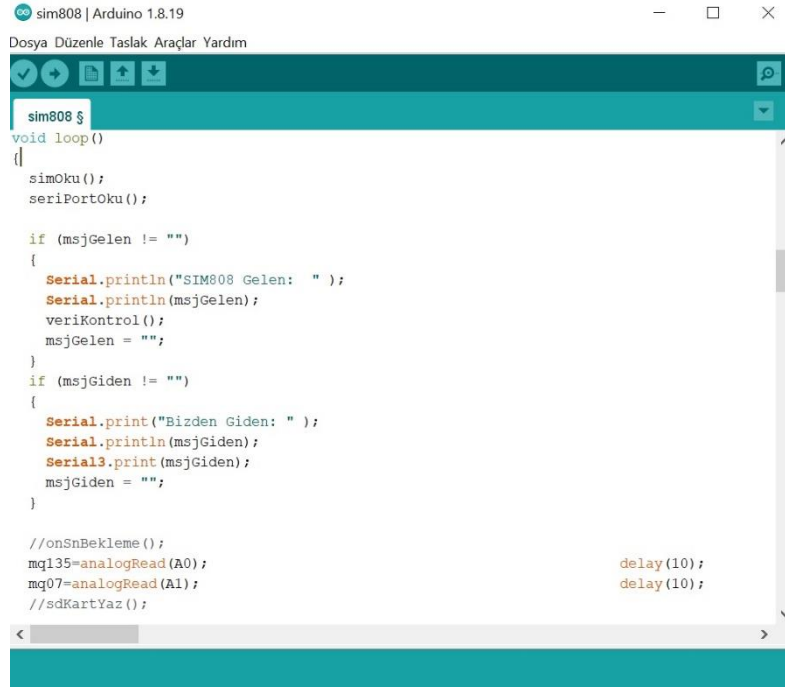
SIM808'den GPS verilerin alınması kısmında, setup() metodunun içinde sim808\_Baslangic() metodu çağrılarak SIM808 için gerekli AT komutların ayarlandı. Kullanılan AT komutlarının ve işlevleri tablo 3.1'de verilmiştir.

**Tablo 3.1.** SIM808 AT Komutları

AT+CGNSPWR=1	Bu komut GNSS için güç kaynağını aktif hale getirir.
AT+CGNSSEQ=RMC	Bu komut GPS yardımı ile gelen veriyi RMC formatında yazar. RMC formatı Saat, Tarih, Konum, Rota ve Hız verileri şeklindedir.
AT+CMGF=1	Bu komut SMS'i text modunda yazmayı sağlar.
AT+CNMI	Bu komut gelen SMS'i seri haberleşme portlarına aktarır.
AT+CMEE=2	Bu komut yazılan kodda hata var ise seri haberleşme kısmından hatanın nedenini aktarır.
AT+W	Bu komut yapılan değişiklikleri SIM'e kaydeder.
AT+CGNSINF	Bu komut GPS'den gelen navigasyon bilgilerini aktarır.
AT+CMGS	Bu komut SMS göndermeyi sağlar.

AT komutlarını ayarladıktan sonra şekil 3.1'de görüldüğü üzere sıra loop() metoduna gelir ve burada simoku() ve seriPortOku() metodları sırasıyla çalışır. SimOku() metodu ile SIM808'den gelen veri kontrol edilir ve msjGelen değişkenine kaydedilir. SimOku() metodunda olası kesintiler için 200 ms döngü eklendi ve bunun için millis() metodu kullanıldı. Bu metot

çağrıldığı sırada Arduino'nun ne zamandır çalıştığını milisaniye cinsinden döner. Serial3.flush() fonksiyonu verinin tamamı gönderilene kadar beklemek için kullanılır. SeriPortOku() metodunda ise seri haberleşme mevcut ise Arduino'dan giden veri okunup msjGiden değişkenine kaydedilir. Aynı şekilde flush() fonksiyonu verinin tamamı gönderilene kadar beklemek için kullanılır. Şekil 3.1'de loop() metodunun içeriği ve şekil 3.2'de simOku() ve seriPortOku() metotları görülmektedir.



```
sim808 | Arduino 1.8.19
Dosya Düzenle Taslak Araçlar Yardım

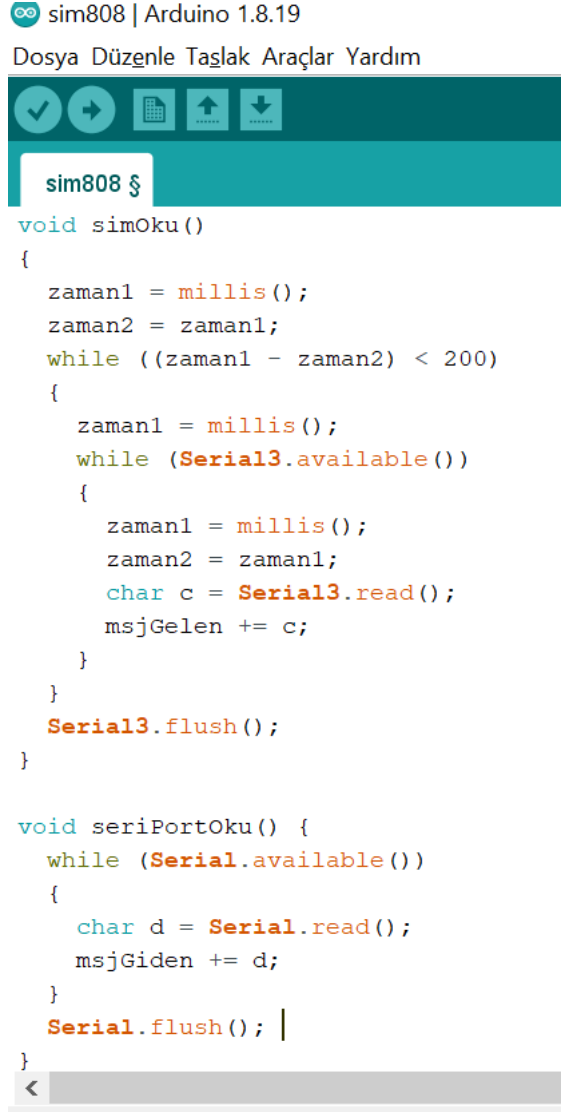
sim808 §
void loop()
{
  simOku();
  seriPortOku();

  if (msjGelen != "")
  {
    Serial.println("SIM808 Gelen: ");
    Serial.println(msjGelen);
    veriKontrol();
    msjGelen = "";
  }
  if (msjGiden != "")
  {
    Serial.print("Bizden Giden: ");
    Serial.println(msjGiden);
    Serial3.print(msjGiden);
    msjGiden = "";
  }

  //onSnBekleme();
  mq135=analogRead(A0);
  mq07=analogRead(A1);
  //sdKartYaz();

  delay(10);
  delay(10);
}
```

**Şekil 3.1.** Analog Pinlerin Tanımlanması



```
sim808 | Arduino 1.8.19
Dosya Düzenle Taşlak Araçlar Yardım

sim808 $
void simOku()
{
    zaman1 = millis();
    zaman2 = zaman1;
    while ((zaman1 - zaman2) < 200)
    {
        zaman1 = millis();
        while (Serial3.available())
        {
            zaman1 = millis();
            zaman2 = zaman1;
            char c = Serial3.read();
            msjGelen += c;
        }
    }
    Serial3.flush();
}

void seriPortOku() {
    while (Serial.available())
    {
        char d = Serial.read();
        msjGiden += d;
    }
    Serial.flush(); |
<
```

**Şekil 3.2.** Arduino ile SIM808 Haberleşme Metodları

Loop() metodundan sırasıyla simOku() ve seriPortOku() metotları çalıştıktan sonra eğer gelen mesaj boş değilse veriKontrol() metoduyla mesajın içeriğini kontrol edilir. Veri kontrol edildikten sonra msjGonder() metodu çalışır ve eğer konumBul() metodu aktif ise mesajı gönderir. VeriKontrol() ve msjGonder() metotları başlık 3.4’de detaylı açıklanacaktır.

KonumBul() metodu şekil 3.3’de görüldüğü gibi SIM808’den gelen mesajı, GPS aktif ve konum bulunduysa, gelen veriyi ayrı ayrı parametrelere bölüp tanımlanan değişkenlere atar. Bunu şekil 3.5’de görülen msjBolme() metodu yapmaktadır. Bu metoda parametre olarak string türünde bir mesaj, char türünde bir ayırıcı ve int türünde bir index gönderilir. Bu metot gelen mesajı, belirtilen ayırıcıya göre bölmektedir. Ardından gönderilen index parametresine göre ayrılan elemanı geri döndürür. AT+CGNSINF metodu ile RMC formatında aldığımız GPS verisinde her

bir parametre virgöl ile bölündüğünden bu metoda virgöl karakteri ayırıcı olarak gönderilir. Böylece veriler gönderilecek mesaj için veya verilerin SD karta kaydedilmesi için hazır hale getirilir. Ayrıca konumBul() metodu konumun bulunup bulunmadığını boolean değişken ile geri döndürür. Böylece mesaj gönderileceği zaman veya veri toplanacağı zaman konumun bulunup bulunamadığı anlaşılabacaktır.

```

sim808 | Arduino 1.8.19
Dosya Düzenle Tağlak Araçlar Yardım

sim808 §
bool konumBul()
{
    Serial.println("GPS Kontrol Ediliyor");
    Serial3.println("AT+CGNSINF");
    simOku();
    delay(1);

    if (msjGelen != "")
    {
        Serial.println("Konum SIM808 Gelen: ");
        Serial.println(msjGelen);
    }

    if ((msjGelen.indexOf("CGNSINF: 1,1,") > -1))
    {
        Serial.println("GPS Aktif Konum Bulundu:");
        tarih = msjBolme(msjGelen, ',', 2);
        yil = tarih.substring(0, 4);
        ay = tarih.substring(4, 6);
        gun = tarih.substring(6, 8);
        saat = tarih.substring(8, 10);
        dakika = tarih.substring(10, 12);
    }
}

```

**Şekil 3.3.** Arduino Konum Bulma Metodu

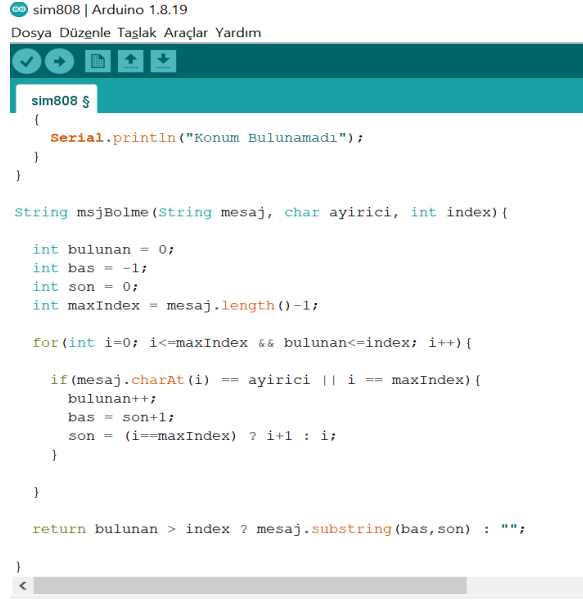
```

sim808 | Arduino 1.8.19
Dosya Düzenle Tağlak Araçlar Yardım

sim808 §
    enlem = msjBolme(msjGelen, ',', 3);
    boylam = msjBolme(msjGelen, ',', 4);
    msjGelen = "";
    return true;
}
else if ((msjGelen.indexOf("CGNSINF: 1,0,") > -1)) //gps den veri gönderiyor
{
    Serial.println("GPS Aktif Konum Bulunamadi Lutfen Bekleyiniz...");
    msjGelen = "";
    return false;
}
else if ((msjGelen.indexOf("CGNSINF: 0,") > -1))
{
    Serial.println("GPS Kapali Acilacak...");
    msjGelen = "";
    return false;
}
else
{
    msjGelen = "";
    return false;
}
}

```

**Şekil 3.4.** Arduino Konum Bulma Metodunun Devamı



```
sim808 | Arduino 1.8.19
Dosya Düzenle Taglar Araçlar Yardım

sim808 $
{
    Serial.println("Konum Bulunamadı");
}

String msjBolme(String mesaj, char ayirici, int index){

    int bulunan = 0;
    int bas = -1;
    int son = 0;
    int maxIndex = mesaj.length()-1;

    for(int i=0; i<=maxIndex && bulunan<=index; i++){

        if(mesaj.charAt(i) == ayirici || i == maxIndex){
            bulunan++;
            bas = son+1;
            son = (i==maxIndex) ? i+1 : i;
        }

    }

    return bulunan > index ? mesaj.substring(bas,son) : "";
}
```

**Şekil 3.5.** Arduino’da GPS Verisini Bölen Metod

Bu aşamada SIM808’de AT komutları kullanılarak GPS’den alınan verilerin, veri toplanacağı zaman veya SMS gönderileceği zaman kullanılması için hazır hale getirilmesi amaçlandı ve başarıyla tamamlandı.

### 3.2. Arduino ile Gaz Sensörlerinin Bağlantısı

Projenin konusu olan Astım hastalarının hangi gazlardan etkilendiğini araştırdıktan sonra donanımsal gereksinimler belirlendi ve bu donanımsal ihtiyaçlar temin edildi. Bunlar: MQ-135 ve MQ-07’dir. Bu sensörler hakkında başlık 2.2’de detaylı bir şekilde açıklama yapıldı. Gelelim bu gaz sensörlerinin Arduino ile donanımsal bağlantısına. Öncelikle MQ-135 için sensörün A0 pini Arduino da A0 pinine bağlandı. Dijital pini kullanmamıza gerek olmadığı için bağlantısı yapılmadı. Sensörün VCC pini Arduino’nun 5V pinine ve sensörün GND pini Arduino’nun GND pinine bağlandı. Böylece donanım kullanıma hazır hale geldi. Aynı işlemler MQ-07 içinde gerçekleştirildi, tek fark sensörün analog pini Arduinoda A1 pinine bağlandı. Arduino IDE programında yazılım bağlantıları için ise mq135 ve mq07 olarak iki ayrı string tanımladı. Başlık 3.1’de ki şekil 3.1’de görüldüğü gibi loop metodu içinde analog pinlerden gelen veriler okundu ve tanımlı olduğu string ifadelerle geri döndürüldü.

### 3.3. Okunan Verilerin SD Karta Yazılması

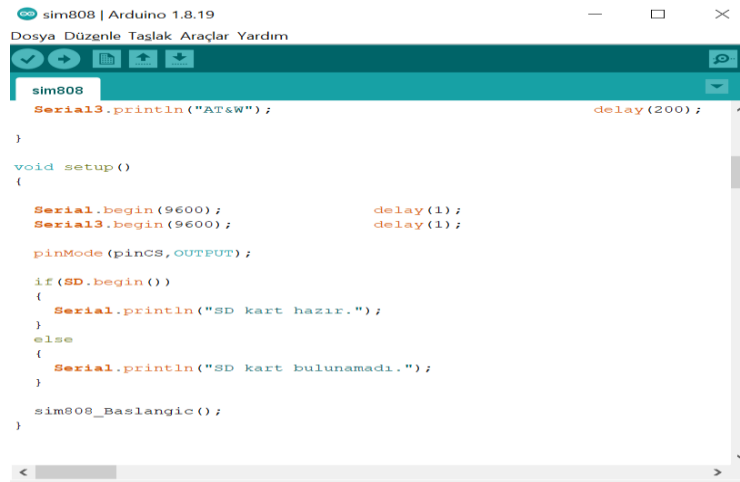
SIM808’den ve gaz sensörlerinden aldığımız verileri kaydetmek için bir SD kart ve SD Kart Modülü temin edildi. SD kart modülünün 6 adet pini vardır. SD kart modülünün Arduino ile

fiziksel bağlantısı tablo 3.2’de verilmiştir. Mikro SD kart modülü SPI haberleşme protokolünü kullandığından Arduino’daki SPI haberleşme pinlerine bağlantısı yapılır. Böylece SD kart modülünün Arduino ile fiziksel bağlantısı tamamlanmaktadır.

**Tablo 3.2.** SD Kart Modülü ile Arduino Mega Bağlantısı

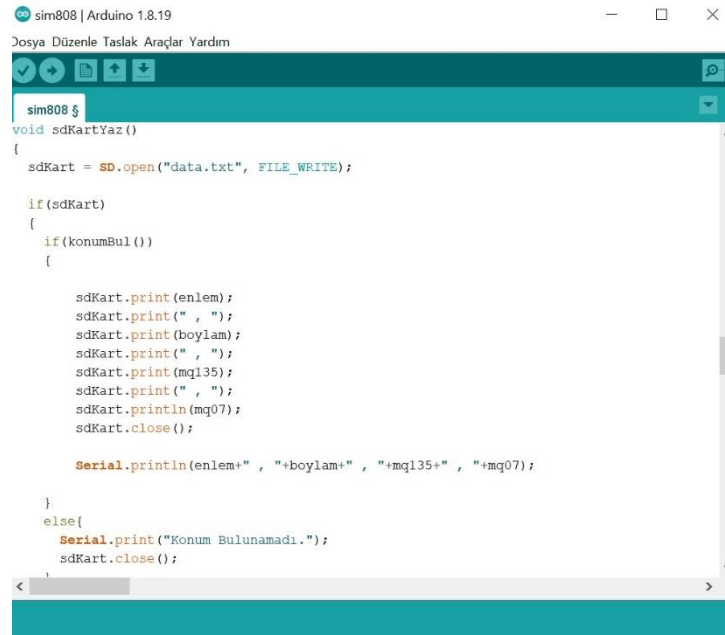
SD Kart Modülü	Arduino Mega
VCC	5V
GND	GND
MISO	Dijital 50
MOSI	Dijital 51
SCK	Dijital 52
CS	Dijital 53

Kod kısmında ise öncelikle SD ve SPI kütüphanelerini projeye eklemek için `#include <SPI.h>`, `#include <SD.h>` kodları eklenir. `pinMode()` metodu ile 53. pin çıkış pini olarak ayarlanır. Şekil 3.2’da gösterildiği üzere `SD.begin()` metodu kullanılarak SD kart ile iletişim başlatılır, eğer SD kart başlamaya hazır ise ekrana “SD Kart hazır” yazılır ve bir sonraki adıma geçer, değilse “SD Kart Bulunamadı” yazdırılır.



**Şekil 3.6.** Arduino Setup Metodu

Sonrasında SD.open() fonksiyonu data.txt metin dosyasını açacaktır. FILE\_WRITE parametresi dosyayı okuma-yazma modunda açar. Dosya açıldıysa ve konumBul() fonksiyonu aktif bir konum bulduysa SD karta sdKart.print() fonksiyonu ile veri yazma işlemi başlayacaktır. Sırasıyla enlem, boylam ve mq135 ile mq07'den gelen veriler SD karta yazılacak ve close() fonksiyonu ile verilerin kaydedilmesi sağlanacaktır. Eğer konum bulunamaz ise ekrana “Konum Bulunamadı” uyarısı verilir ve aynı şekilde SD kart kaydedilerek kapatılır. Dosyaya hiç ulaşamaz ise ekrana “Dosya Açılmadı” uyarısı verilir.



Şekil 3.7. Arduino’da SD Karta Yazma Metodu

### 3.4. SIM808 ile SMS Gönderme

SIM808 ile SMS gönderme işlemine gelen verileri kontrol etmek amacıyla veriKontrol() metodu yazılarak başlandı. Gelen mesaj bilgi sözcüğü ise bilgi almak üzere SMS gönderildiği anlaşılır. Daha önce AT komutları yardımı ile SMS’in seri haberleşme portlarından aktarılması sağlanmıştı. SMS içeriği başında gönderen telefon numarası ile Arduino’ya gönderilmekte. Bu içerik substring() metodu yardımı ile bölündü ve gönderen telefon numarası elde edildi. Daha sonra SMS’i gönderen kişiye msjGonder() metoduyla SMS gönderildi.



```
sim808 | Arduino 1.8.19
Dosya D zenle Tařlak Aralar Yardım

void veriKontrol()
{
  if (( (msjGelen.indexOf("bilgi")) > -1 || (msjGelen.indexOf("Bilgi")) > -1 || (msjGelen.indexOf("BILGI")) > -1 ))
  {
    telefonNo = msjGelen.substring(12,22);
    Serial.println("telefonNO:"+telefonNo);
    msjGelen = "";
    return msjGonder();
  }
}
```

řekil 3.8 Arduino’da Gelen SMS’in Kontrol 

Gelen mesaj bilgi s zc ğ n  ieriyorsa veriKontrol() metodundan msjGonder() metoduna geri d n ř yapıldı. MsjGonder() metodunda konum kontrol edildi, konum aktif ise GPS’in uyduda baėlandığı saat dilimi farklı olduėundan, konumBul() metodunda alınan saat deėiřkeni    ile toplandı. Daha sonra AT+CMGS komutuyla ve kullanıcından alınan telefon numarasıyla SMS g nderme iřlemi bařlatıldı. AT+CMGS komutunun ardından char 26 deėerine kadar SMS ieriėi yazıldı. SIM808 char 26 deėeri ile SMS ieriėinin sonlandığını anlamakta, b ylece SMS g nderilmiř oldu. Kod sonunda verinin tamamı g nderilene kadar beklemek iin flush() fonksiyonu eklendi.



```
sim808 | Arduino 1.8.19
Dosya D zenle Tařlak Aralar Yardım

void msjGonder()
{
  if (konumBul())
  {
    int s = saat.toInt()+3;
    saat = String(s);

    Serial3.print("AT+CMGS="+s); //sms g ndermeyi saėlayan komut
    Serial3.print(telefonNo);
    Serial3.print("\r\n");
    Serial3.println("Platforma hosgeldiniz, saėliginiz icin elimizden geleni yapiyor, yuksek dogrulukla verileri sizlerle paylasiyoruz.Marmarisde astim hastalari icin dikk
    Serial3.println("Dikkatli Olunmasi Gereken Yerler:");
    Serial3.println("Pazar Yeri, Sanayi ");
    Serial3.println("Tehlikeli Yerler: ");
    Serial3.println("Sariana ");
    Serial3.println("Tarih : " + gun + "/" + ay + "/" + yıl);
    Serial3.println("Saat : " + saat + ":" + dakika);
    Serial3.println((char)26);
    Serial3.flush();

  }
  else
  {
    Serial.println("Konum Bulunamadı");
  }
}
```

řekil 3.9. Arduino’da SMS G nderen Metot



### 3.5. HTML ile Web Sitesi Tasarımı

Projenin bu aşaması Web bölümünün ilk aşamasıdır. Bir web sitesi oluşturmak için öncelikle HTML yapısı oluşturulmalıdır. Bunun için öncelikle tasarımın nasıl olacağı kararlaştırıldı. Web sitesinin en üstünde linklerin bulunacağı bir bar, sonrasında ise sırası ile astım hakkında kısa bir bilgi, üzerinde verilerin işlendiği Google Map ve son olarak da mail almak isteyen kullanıcılar için abonelik formu. Daha sonra kullanılacak logo ve fontlar seçildi. Google font aracılığı ile yazı fontları, Font Awesome aracılığı ile de logo seçildi. Ardından bütün sayfa tek bir div içerisinde yazılmaya başlandı. İlk olarak sitenin en üst bölümünde gözükecek logo ve link bar kısmı yazıldı. Sitenin başlangıç ve link bölümü olduğundan header ve nav etiketleri arasında yazıldı. Şekil 3.10'da eklenen logo ve fontların ayrıca header kısmının kodlarını görmektesiniz.

```
deneye > index.php
1  <!DOCTYPE html>
2
3  <html lang="en">
4
5  <head>
6
7      <meta charset="UTF-8">
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10     <link href="https://fonts.googleapis.com/css2?family=M+PLUS+1+Code&display=swap" rel="stylesheet">
11     <link href="https://fonts.googleapis.com/css2?family=Lobster&display=swap" rel="stylesheet">
12
13     <script src="https://kit.fontawesome.com/c20485228a.js" crossorigin="anonymous"></script>
14
15     <title>Astım</title>
16
17 </head>
18
19 <body>
20
21     <header class="header">
22         <nav>
23             <a href="#" class="logo">
24                 <i class="fas fa-heartbeat"></i>
25             </a>
26
27             <ul>
28                 <li><a href="#astim-bilgi"> Astım Nedir</a></li>
29                 <li><a href="#map-bilgi"> Haritalar</a></li>
30                 <li><a href="#abone"> Abonelik</a></li>
31             </ul>
32         </nav>
33     </header>
34
35 </body>
36
37 </html>
```

Şekil 3.10. Kurulan HTML Yapısı ve Header Bölümü

Daha sonra div etiketleri kullanılarak astım ve harita bölümleri oluşturuldu. Astım kısmında içerik olarak bir h1 etiketi, bir h2 etiketi ve iki p etiketi kullanılarak astım ve astımı olumsuz etkileyen gazlar hakkında bilgi verildi. Şekil 3.11'de astım kısmının kodlarını görmektesiniz. Harita bölümünde ise başlık için bir h1 etiketi, haritanın JavaScript bölümünde

yerleştirilmesi için bir div etiketi, ayrıca harita üzerinde işlenen bölgelerde hangi rengin ne kadar riskli olduğunun açıklandığı bir başka div etiketi yazıldı. Bunu da şekil 3.12’de görmekteyiz.

```
<div id="astim-bilgi">

  <h1 class="astim-baslik">Astım Nedir?</h1>
  <p class="astim-text">
    Astım, solunum yollarını tıkayan ve nefes almayı zorlaştıran kronik bir hastalıktır. Astımda, hava yolu etrafındaki düz kaslar,
    alerjenler, duman, hava kirliliği, soğuk hava veya egzersiz gibi tetikleyicilere yanıt olarak kasılır ve mukus adı verilen yapışkan
    salgının üretimi artar; bu durum hava yolunun daralmasına neden olur.
  </p>
  <h2 class="astim-yanbaslik">Astım'ı Olumsuz Etkileyen Gazlar</h2>
  <p class="astim-text">
    Tetikleyici faktörler hava yolu inflamasyonu yerleşmiş astımlı hastalarda akut
    ataklara neden olan faktörlerdir. Soba ve fırın yakıtları, kızarmış yağlar, oda
    spreyleri, boya ve ciladan kaynaklanan azot oksit, karbon monoksit, karbon dioksit,
    sülfür dioksit ve formaldehit gibi gazlar evici hava kirliliği nedenleridir.
    Isınma ve sanayide kullanılan yakıtlardan ve egzozlardan açığa çıkan sülfürdioksit, ozon, azot oksitler, asit aerosoller
    ve partiküller dış ortam hava kirliliğine neden olurlar.
  </p>
</div>
```

**Şekil 3.11.** Astım Bilgi Bölümünün HTML Yapısı

```
<div id="map-bilgi">

  <h1 class="map-baslik">Tehlikeli Bölgeler</h1>

  <div id="map"></div>

  <div class="map-text">

    <p class="kirmizi">
      Kırmızı Bölgeler : Tehlikeli Bölgeler
    </p>

    <p class="turuncu">
      Turuncu Bölgeler : Dikkatli Olunması Gereken Bölgeler
    </p>

    <p class="yesil">
      Yeşil Bölgeler : Temiz Havaya Sahip Bölgeler
    </p>

  </div>
</div>
```

**Şekil 3.12.** Harita Bölümünün HTML Yapısı

HTML bölümünün son kısmı olan abonelik formunda ise içerik olarak yine başlık için bir h1 etiketi, formun etkileşime girebilmesi için form etiketi, her bir form elemanı için olmak üzere bir div etiketi içerisinde label ve input etiketleri kullanıldı. Şekil 3.13’de bu bölümün kodlarını görmekteyiz.

```

<div id="abone">
  <h1 class="abone-baslik">Abonelik</h1>
  <form action="" method="post">
    <div class="form-group">
      <label for="isim">Ad Soyad:</label>
      <input class="girdi" type="text" id="isim" name="be_isim" placeholder="Ad Soyad Giriniz...">
    </div>
    <div class="form-group">
      <label for="sehir">Şehir:</label>
      <input class="girdi" type="text" id="sehir" name="be_sehir" placeholder="Yaşadığınız Şehri Giriniz...">
    </div>
    <div class="form-group">
      <label for="tel-no">Telefon Numarası:</label>
      <input class="girdi" type="text" id="tel-no" name="be_tel" placeholder="Telefon Numaranızı Giriniz...">
    </div>
    <div class="form-group">
      <label for="mail-input">E-Posta:</label>
      <input class="girdi" type="text" id="mail-input" name="be_mail" placeholder="E-Posta Adresinizi Giriniz...">
    </div>
    <button type="submit" class="btn">
      Gönder
    </button>
  </form>
</div>

```

**Şekil 3.13.** Abonelik Bölümünün HTML Yapısı

Bu kısımda yazıldıktan sonra footer etiketleri kullanılarak sayfanın sonuna proje sahiplerinin isimleri ve proje yılı yazıldı. Böylece projenin HTML bölümü tamamlanmış oldu. Bu süreç içerisinde yazılan HTML kodlarında class ve id atamaları yapıldı. Header bölümünde bulunan a etiketlerinin href özelliklerine ise her bir bölüm divinin idsi yazıldı. Böylece linklere tıklandığında ilgili bölüme gidilmesi sağlandı. Bundan sonraki süreçte CSS ve JavaScript bölümünde ise bu id ve class yapıları kullanıldı.

### 3.6. CSS ile Web Sitesi Tasarımı

CSS bölümünde web sitesinin daha rahat anlaşılabilir ve güzel bir tasarıma sahip olması sağlandı. Farklı ekran çözünürlüklerinde iken farklı boyutlarda çalışabilmesi için responsive bir tasarım düşünüldü. Proje içerisinde sık kullanılan CSS kodlarının işlevleri tablo 3.3’de görülmekte. İlk olarak genel html yapısının font büyüklükleri, sitenin arka planı ve container çerçevesi ayarlandı. Container sınıfına belirli bir genişlik verilerek sayfanın sağdan ve soldan boşluklu olması sağlandı. Şekil 3.14’de bunlar görülmekte.

```

denet > style.css > $html
1  *{
2      margin: 0;
3      padding: 0;
4      box-sizing: border-box;
5  }
6
7  html{
8      font-size: 10px;
9  }
10
11 body{
12
13     background-image:linear-gradient(to left bottom, rgba(230,255,230,0.3), rgba(230,255,230,0.3)),
14     url(img/bg4.jpg);
15     background-size: cover;
16     background-position: left;
17     background-attachment: fixed;
18 }
19
20
21 .container{
22     width: 110rem;
23     margin-right: auto;
24     margin-left: auto;
25 }
--

```

**Şekil 3.14.** CSS ile HTML ve Body Etiketlerinin Tasarımı

**Tablo 3.3.** Sık Kullanılan CSS Kodları

Display	Bir html etiketinin ekranda nasıl yer kapladığının belirtildiği alandır. Block ise satırı kapsar, inline ise içerik kadar yer kaplar, inline-block ise içerik kadar kaplar fakat block eleman gibi davranabilir, son olarak da flex ise daha esnek bir şekilde konumlandırma sağlar.
Padding	Bir html etiketi eğer ki display inline değil ise iç boşluk vermeyi sağlar. Inline etiketlere padding verilemez. İstenilen kenardan boşluk verilebilir.
Margin	Bir html etiketine dış boşluk vermeyi sağlar. Padding gibi marginde de istenilen kenardan boşluk verilebilir.
Font-Size	Etiket içerisinde bulunan font boyutlarının ayarlanmasını sağlar.
Font-Family	Seçilen etiketin font türünün değiştirilmesini sağlar.
Background-Color	Arka plana renk vermeyi sağlar.
Color	Color ise içeriğe renk vermeyi sağlar.
Height	Seçilen html etiketine istenilen bir uzunluğun verilmesini sağlar.
Width	Seçilen html etiketine istenilen bir genişliğin verilmesini sağlar.
Text-Align	Seçilen metnin sol, sağ veya merkezde konumlandırılmasını sağlar.
Border-Radius	Çerçeve köşelerinin yuvarlaklaştırılmasını sağlar.

Container ve arka plan ayarlandıktan sonra sayfanın en üst bölgesi olan header bölümüne geçildi. Header etiketine direk ulaşarak sağ alt ve sol alt köşelere 10px radius verildi. Daha sonra sırası ile ul, li, a etiketleri ve logo classı için margin, padding gibi boşluk ayarlamaları yapılmasının yanı sıra, text-decoration, list-style-type gibi tasarımlar kaldırıldı. Bu değişiklikleri şekil 3.15’da görmekteyiz.

```
29  .header{
30      background-color: whitesmoke;
31      height: 4rem;
32      border-bottom-left-radius: 10px;
33      border-bottom-right-radius: 10px;
34      margin-bottom: 4rem;
35  }
36
37  ul{
38      font-family: 'Lobster', cursive;
39      list-style-type: none;
40      float: right;
41  }
42
43  li{
44      display: inline-block;
45      font-size: 2rem;
46      margin-top: 5px;
47      margin-right: 2rem;
48  }
49
50  a{
51      text-decoration: none;
52      color: crimson;
53  }
54
55  .logo{
56      margin-left: 3rem;
57      padding: 1rem;
58      font-size: 3.5rem;
59  }
```

**Şekil 3.15.** CSS ile Header Bölümünün Tasarımı

Daha sonra ise astım hakkında genel ve kısa bir bilginin verileceği bölümün tasarımına geçildi. Tasarıma göre dört köşesi ovalleştirilmiş bir görüntü ekrana çıkartıldı. İçeriğe göre bir yükseklik verilerek, başlıklar ve paragraflar için gerekli kodlar yazıldı. Aynı zamanda bu astım bölümünde de bir arka plan resmi kullanıldı. Bununla beraber tabii ki margin ve padding kullanılarak boşluklar ayarlandı.

```

65  #astim-bilgi{
66      background-image:linear-gradient(□rgba(255,255,255,0.7),□rgba(255,255,255,0.7)),
67      url(img/astim.jpg);
68
69      border-radius: 50px;
70      background-size: cover;
71      padding: 5rem;
72      height: 75vh;
73      margin-bottom: 5rem;
74  }
75
76  .astim-baslik{
77      margin-bottom: 2rem;
78      text-align: center;
79      font-family: 'M PLUS 1 Code', sans-serif;
80      font-size: 7rem;
81  }
82
83  .astim-text{
84      font-size: 2rem;
85      font-family: 'M PLUS 1 Code', sans-serif;
86  }
87
88  .astim-yanbaslik{
89      margin: 2rem 0;
90      font-size: 4rem;
91      font-family: 'M PLUS 1 Code', sans-serif;
92  }

```

**Şekil 3.16.** CSS ile Astım Bilgi Bölümünün Tasarımı

Harita kısmında ise bilgilendirme metinleri için yeşil, turuncu ve kırmızı sınıfları ile renkler ayarlandı. HTML bölümünde boş bırakılan map divinin yükseklik ve genişlikleri ayarlandı. Bunun dışında astım bilgi bölümü ile aynı yapı oluşturuldu.

```

97  #map-bilgi{
98      border-radius: 50px;
99      width: 100%;
100     height: 80vh;
101     background-color: □whitesmoke;
102     margin-bottom: 5rem;
103 }
104
105 .map-baslik{
106     padding-top: 1.5rem;
107     margin-bottom: 2rem;
108     text-align: center;
109     font-family: 'M PLUS 1 Code', sans-serif;
110     font-size: 5rem;
111 }
112
113 #map{
114     margin-left: 10rem;
115     margin-bottom: 2rem;
116     border-radius: 20px;
117     width: 80%;
118     height: 60%;
119 }
120
121 .map-text{
122     margin-left: 10rem;
123     font-size: 2rem;
124     font-family: 'M PLUS 1 Code', sans-serif;
125 }
126
127 .kirmizi{color: □tomato; margin-bottom: 5px;}
128
129 .turuncu{color: □orange; margin-bottom: 5px;}
130
131 .yesil{color: ■green;}

```

**Şekil 3.17.** CSS ile Harita Bölümünün Tasarımı

Son bölüm olan abonelik formu bölümünde ise input ve label etiketleri içeren formun düzgün bir görüntüye sahip olması sağlandı. Forumun gönderilmesi için eklenen butonun renk, boyut ve konum tasarımı yapıldı. Bunun haricinde butona hover olayı eklendi. Hover sayesinde

butonun üstüne gelindiğinde arka plan ve içerik rengi değişmekte. Ayrıca cursor poniter olarak ayarlandığı için fare imleci de değişmekte. Şekil 3.18’de abonelik forumu için yazılan CSS kodları görülmekte. Şekil 3.19’da ise mesaj sınıfı ile mail gönderimi başarılı olunca çıkan mesajın konumları ayarlandı. Ayrıca en son footer etiketine ulaşarak sayfa sonundaki içeriğin font büyüklüğü ayarlandı ve ortalandı.

```
135 #abone{
136     border-radius: 50px;
137     width: 100%;
138     height: 60vh;
139     background-color: #whitesmoke;
140     margin-bottom: 5rem;
141 }
142
143 .abone-baslik{
144     padding-top: 1.5rem;
145     margin-bottom: 2rem;
146     text-align: center;
147     font-family: 'M PLUS 1 Code', sans-serif;
148     font-size: 5rem;
149 }
150
151 .form-group{
152     margin-left: 8rem;
153     margin-bottom: 2rem;
154     font-size: 2rem;
155     color: #crimson;
156     font-family: 'M PLUS 1 Code', sans-serif;
157 }
158
159 .girdi{
160     width: 70%;
161     padding: 1rem;
162     border: 1px solid #thistle;
163 }
164
165 #isim{margin-left:7.6rem;}
166
167 #sehir{margin-left: 10.2rem;}
168
169 #mail-input{margin-left: 8.5rem;}
```

**Şekil 3.18.** CSS ile Abonelik Formunun Tasarımı

```

165  < .btn{
166      margin-left: 25.5rem;
167      margin-bottom: 2rem;
168      padding: 1rem;
169      font-size: 2rem;
170      font-family: 'M PLUS 1 Code', sans-serif;
171      background-color: whitesmoke;
172      border: 1px solid darkorchid;
173      color: darkorchid;
174      border-radius: 5px;
175      cursor: pointer;
176  }
177
178  < .btn:hover{
179      background-color: darkorchid;
180      color: whitesmoke;
181  }
182  }
183
184  < .mesaj{
185      font-size: 10px;
186      margin-left: 8rem;
187  }
188
189  < footer{
190      text-align: center;
191      font-size: 2rem;
192      font-family: 'Lobster', cursive;
193      margin-bottom: 1rem;
194  }

```

**Şekil 3.19.** CSS ile Buton ve Footer Tasarımı

Projenin CSS kısmında son olarak responsive tasarımı yapıldı. Başlangıçta proje geniş ekran çözünürlüğüne göre yazıldı. Bu bölümde ise daha düşük çözünürlükte gerekli olan değişiklikler yapıldı. İlk olarak genişlik 1150px'e geldiğinde html seçicisi ile yazılan font-size düşürüldü. Bütün proje bu değere endeksli yazıldığından direkt proje üzerinde genel olarak font ve boşluk büyüklükleri değişmiş oldu. Aynı şekilde çözünürlük 950px'e düştüğü zaman da font-size düşürüldü. Son olarak 750px için bir responsive tasarım yapıldı. Fakat bu çözünürlükte font-size fazla küçüldüğünden tasarım kötüleşti. Bunun için belli başlı düzeltmeler ile daha iyi görünen bir görüntü elde edildi. Yapılan düzeltmeler Şekil 3.21'de görülmektedir. Sonuç Olarak CSS bölümünde, web sayfası düzgün bir tasarım ile buluşmuş oldu.

```

205  < @media (max-width: 1150px){
206      |
207      |     html{ font-size: 8px; }
208      |
209      | }
210
211  < @media (max-width: 950px){
212      |
213      |     html{ font-size: 6.5px; }
214      |
215      |     #astim-bilgi{ height: 55vh; }
216      |
217      |     #map-bilgi{ height: 55vh; }
218      |
219      |     #abone{ height: 45vh; }
220      |
221      | }

```

**Şekil 3.20.** CSS ile Responsive Tasarım



```

223  √ @media (max-width: 750px){
224
225      html{ font-size: 4.5px; }
226
227      .header{ height: 6rem; }
228
229      li{ font-size: 4rem; }
230
231      .logo{ font-size: 5rem; }
232
233      #astim-bilgi{ height: 55vh; }
234
235      .astim-text{ font-size: 3rem; }
236
237      #map-bilgi{ height: 50vh; }
238
239      .map-text{ font-size: 3rem; }
240
241      #abone{ height: 40vh; }
242
243      .form-group{ font-size: 3rem; }
244
245      #isim{ margin-left: 11.4rem; }
246
247      #sehir{ margin-left: 15.3rem; }
248
249      #mail-input{ margin-left: 12.8rem; }
250
251  √ .btn{
252      |     margin-left: 34rem;
253      |     font-size: 3rem;
254      | }

```

**Şekil 3.21.** CSS ile 750 px İçin Responsive Tasarım

### 3.7. JavaScript ile Google Map API Kullanımı

Öncelikle Google Cloud Platforma giriş yapıldı, ardından yeni bir proje oluşturup kimlik bilgileri kısmından API key alındı. Google Map API bu API key aracılığı ile projeler üzerinde kullanılmaktadır. API, alınan bu key ile beraber HTML dosyasına script olarak eklendi bu sayede site üzerinde kullanılabiliyor. Aynı zamanda kullanacağımız JavaScript dosyası da script olarak eklendi. Böylece HTML eklemeleri bitti bundan sonra JavaScript kısmı başladı.

Arduino bölümünde alınıp sd karta kaydedilen hava kirliliği verileri, JavaScript üzerinde her bir bölge için obje olarak eklendi. Her bir objede enlem boylamın tutulduğu bir başka obje ve bölgenin hangi risk analizi grubunda olduğu yazıldı. Grupların numaralandırmaları yeşil, turuncu, kırmızı sırası ile yapıldı. Daha sonra bir forEach döngüsü ile çizilmek üzere bütün ojeler bir dizide saklandı. Bu yapı şekil 3.22’de görülmekte.

```

1  bolge1={
2  |   merkez:{lat: 36.856742, lng: 28.274592},
3  |   grup: 3,
4  | };
5
6  bolge2={
7  |   merkez:{lat: 36.856247, lng: 28.266437},
8  |   grup: 2,
9  | };
10
11 bolge3={
12 |   merkez:{lat: 36.858940, lng: 28.255053},
13 |   grup: 1,
14 | };
15
16 bolge4={
17 |   merkez:{lat: 36.864763, lng: 28.269610},
18 |   grup: 2,
19 | };
20
21 bolge5={
22 |   merkez:{lat: 36.863493, lng: 28.275427},
23 |   grup: 1,
24 | };
25
26 var bölgeler = [bolge1, bolge2, bolge3, bolge4, bolge5];
27

```

**Şekil 3.22.** Haritaya Çizilen Veriler

Daha sonra API kullanılarak haritanın çizilmesi yapıldı. Map metoduna haritanın ekleneceği bölüm ve içerisinde harita yüksekliği, ilk açılış konumu, harita tipi verilerinin bulunduğu bir obje gönderildi. Haritaya verilerin işlenmesi için çember çizimi tercih edildi. Oluşturulan diziyi forEach metodu ile gezip her bir bölge için, bulunduğu gruba göre rengi değişen bir çember çizildi. Bunun için circle metodu kullanıldı, bu metoda parametre olarak içerisinde dış çizgi rengi, dış çizgi opaklığı, dış çizgi kalınlığı, iç rengi, iç opaklığı, çizilecek harita, çemberin merkez noktası ve son olarak da yarıçap verilerinin bulunduğu bir obje gönderildi. Bütün bunlar bir fonksiyon içerisinde yazıldı, kodları şekil 3.23’de görülmektedir. Böylece JavaScript ve Google Map API ile verilerin haritaya işlenmesi de tamamlandı.

```

29 function initMap() {
30
31     //MapTypes : roadmap,terrain,hybrid,satellite
32     const map = new google.maps.Map(document.getElementById("map"), {
33         zoom: 15,
34         center: {lat: 36.856742, lng: 28.274592},
35         mapTypeId: "roadmap",
36     });
37
38     var renk;
39
40     bölgeler.forEach(bölge => {
41
42         switch(bölge.grup){
43
44             case 1:
45                 renk = "#00CC00";
46                 break;
47
48             case 2:
49                 renk = "#F07400";
50                 break;
51
52             case 3:
53                 renk = "#FF0000";
54                 break;
55
56             default:
57                 renk = "#FFF";
58                 break;
59         }
60
61         new google.maps.Circle({
62             strokeColor: renk,
63             strokeOpacity: 0.8,
64             strokeWeight: 2,
65             fillColor: renk,
66             fillOpacity: 0.20,
67             map,
68             center: bölge.merkez,
69             radius: 100,
70         });
71     });
72 }
73
74

```







**Şekil 3.23.** Verilerin Haritaya Çizilmesi

Bu bölümde sonuç olarak statik bir veri kümesi ile riskli bölge analizi yapıldı. Elbette ki tam olarak istenileni veren bir sistem yapılamadı. Bunun için şehrin farklı bölgelerinden, olabildiğince yakın mesafelerde, daha kaliteli ve çeşitli sensörler aracılığı ile veriler çekilmeli, üstelik bu sensörler internete bağlanarak verileri veri tabanına kaydedebilmeli. Bu durumda dinamik olarak alınan verilere JavaScript aracılığı ile ulaşılmalı ve harita üzerine aktarılmalıydı. Böylece astım hastaları veya ciğerlerini zararlı gazlardan korumak isteyen bireyler site üzerinden anlık olarak verilere ulaşarak sağlıklarını koruyabilirdi. Fakat böyle bir durum bizim için mali açıdan mümkün olmadığından, projede statik veriler kullanılarak işin yapılabilirliği gösterilmiş oldu.

### 3.8. XAMPP ile Veri Tabanının Oluşturulması

Başlık 2.8’de XAMPP in kurulumundan bahsedildi, şimdi ise program aracılığı ile veri tabanı oluşturuldu. Öncelikle XAMPP üzerinden Apache ve MySQL servisleri start butonu ile başlatıldı. Apache hizmeti ile local host sağlandığından buradan admin butonuna basıldığında host klasörüne eklenmiş projeler açılabilmekte. Bu bölüm MySQL ile veri tabanı oluşturulduğundan oradan admin butonuna basıldı ve karşımıza phpMyAdmin servisi çıktı. Buradan veri tabanı oluşturmaya başlandı. Abone bilgileri ismi ile oluşturulan veri tabanına iki

tane tablo yapısı eklendi. İlk şekil 3.24’de örnek olması için elle doldurulan aboneler tablosu. Bu tabloya web sitesinden abonelik formuna doldurulan bilgiler aktarıldı ve burada kayıt altına alındı. Primary key olarak otomatik artan id kullanıldı, bunun dışında telefon numarası ve email aynı veriden yalnızca bir tane olması için uniq yani tekil olarak ayarlandı. Kullanıcı hakkında bilgi sahibi olunması amacı ile isim ve şehir bilgileri de alındı.

				ID	isim	sehir	tel_no	email
<input type="checkbox"/>				1	Ekrem Bali	Bursa	05320122937	ekrem.bali16@outlook.com
<input type="checkbox"/>				4	Doğukan Aydoğdu	Muğla	05445783119	doguaa09@gmail.com

**Şekil 3.24.** Aboneler Tablosu

Daha sonra ikinci tablo olan konum ve sensör verilerinin tutulduğu tablo oluşturuldu. Bu tabloda yine aynı şekilde bir id, bunun haricinde double veri tutan enlem ve boylam sütunları, int veri tutan mq135, mq07 ve grup sütunları, son olarak ise semt isimlerinin tutulduğu varchar türünde bir sütun kullanıldı. Bu tablo zaten statik olarak doldurulduğundan arduino ile verilerin toplanması aşamasında alınan örnekler ile dolduruldu. Şekil 3.25’de bu tablo görülmektedir.

					id	enlem	boylam	mq135	mq07	grup	semt
<input type="checkbox"/>				Sil	1	36.85675	28.274587	56	100	3	Sarıana
<input type="checkbox"/>				Sil	2	36.85634	28.266408	48	73	2	Pazar Yeri
<input type="checkbox"/>				Sil	3	36.85894	28.255053	27	49	1	Caprice AVM
<input type="checkbox"/>				Sil	4	36.864763	28.26961	68	55	2	Sanayi
<input type="checkbox"/>				Sil	5	36.863493	28.275427	34	57	1	Otogar

**Şekil 3.5.** Konum ve Sensör Verilerinin Tutulduğu Tablo

Bu aşamada yine tamamen webe bağlı bir yapı elde edilmemiştir. XAMPP yardımı ile projenin çalıştığını gösteren local bir sunucu kurulmuş, bunun üzerinde proje çalıştırılmıştır. Normal şartlarda projenin bireylere faydasının dokunması için bir domain alınmalı ve global bir host hizmeti ile web sitesi internete açılmalıdır. Lakin hali hazırda statik veri kümelerinin kullanılmasından dolayı tam işlev kazanmayan projede domain ve host için de masraf yapılmadan local olarak sitenin işlevsellikleri kanıtlanmıştır.

### 3.9. PHP ile Server Bağlantısının Kurulması

Öncelikle yeni bir dosya açılarak veri tabanı ile bağlantı sağlandı. Bu bağlantının sağlanması için sunucu, veri tabanı kullanıcısı ve şifresi, son olarak veri tabanı adı değişkenlere

atandı. Ardından yeni bir mysqli objesi oluşturularak bu değişkenler parametre olarak gönderildi. Bu veri tabanı bağlantısının karakter seti utf8 olarak ayarlandı ve son olarak bağlantının başırlanamaması durumunda, bağlantının öldürölmesi ve bir mesajın gönderilmesi sağlandı. Ana index.php dosyasında, bu baglanti.php dosyası include edildi, böylece veri tabanına bağlantı sağlandı. Şekil 3.26’da bağlantının yapıldığı kodlar görünmektedir.

```
baglanti.php
1  <?php
2
3  $vt_sunucu="127.0.0.1";
4  $vt_kullanici="root";
5  $vt_sifre="";
6  $vt_adi="abonebilgileri";
7
8  //veritabanına bağlanma komutu
9  $baglan = new mysqli($vt_sunucu, $vt_kullanici, $vt_sifre, $vt_adi);
10 $new = mysqli_set_charset($baglan,"utf8");
11
12 //bağlanamadıysak:
13 if($baglan->connect_error){
14     die("Veritabanı bağlantı işlemi başarısız.".$baglan->connect_error);
15 }
16
17 ?>
```

**Şekil 3.26.** Veri Tabanı Bağlantısı

Projenin web kısmının son aşaması olan PHP ile web programlama için öncelikle PHPMailer kütüphanesi projeye eklendi. Bu kütüphane sayesinde bir SMTP sunucu ile bağlantı kurulup mail gönderilmesi sağlandı. Kendi sunucumuz olmadığından kullanımı ücretsiz SMTP sunucularından olan Google ve Microsoft sunucuları kullanıldı. Bu sebepten yalnızca smtp.gmail.com ve smtp.live.com sunucularına bağlı mail hesaplarına mail gönderilebilmekte. Bu bölümde kısaca form üzerinden alınan veriler öncelikle veri tabanına aktarıldı, ardından konum ve sensör verilerinin tutulduğu tablodan dikkatli olunması gereken ve tehlikeli bölgelerin bilgileri alındı. Bu bilgilerden yararlanılarak formda doldurulan mail adresine bu bölgelerin Google haritalardaki konum linkleri semt isimleriyle birlikte yollandı.

İlk olarak projeye PHPMailer kütüphanesinin eklenmesi yapıldı. Bu kütüphane aracılığı ile gelen PHPMailer sınıfından bir nesne oluşturuldu. Daha sonra bu nesne ile proje için gerekli değişkenler ayarlandı. Şekil 3.27’de yazılan kodlar ve tablo 3.4’de de proje boyunca PHPMailer kütüphanesinden kullanılan kodların açıklamaları bulunmaktadır.

**Tablo 3.4.** Projede Kullanılan PHPMailer Değişken ve Metodları

isSMTP()	Mailin SMTP ile gönderilmesini sağlar.
SMTPAuth	SMTP kimlik doğrulayıcının kullanılıp kullanılmayacağını tutan değişken.
Username	Giriş yapılacak mail adresi.
Password	Giriş yapılacak mail adresinin şifresi.
Host	Şifreleme yöntemleri ve portları ile birlikte bağlanılacak SMTP sunucular.
CharSet	Mail içeriğinde kullanılacak karakter seti.
setFrom()	Mailin gönderileceği mail adresinin ve mail başlığının belirlendiği metod.
addAddress()	Maili alacak mail adresi ve isminin belirlendiği metod.
isHTML()	Mail içeriğinde HTML etiketlerinin kullanılıp kullanılmayacağını belirlendiği metod.
Subject	Mail başlığını tutan değişken.
Body	Mail içeriğini tutan değişken.
send()	Mailin gönderilmesini sağlayan metod.

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';

$mail = new PHPMailer();

$mail->isSMTP();

$mail->SMTPAuth = true;
$mail->Username = "bali.ekrem16@gmail.com";
$mail->Password = "1";

$mail->Host = "tls://smtp.gmail.com:587";
$mail->CharSet = 'UTF-8';

$mail->setFrom(address:"bali.ekrem16@gmail.com", name:"Astim-Saglik");
```

**Şekil 3.27.** PHPMailer Kütüphanesinin Eklenmesi ve Mail Ayarları

Mailin gönderileceği adres ve STMP hostları belirlendi. Bu aşamadan sonra ilk olarak formdan alınan verilerin aboneler tablosuna eklenmesi işlemi yapıldı. Bunun için formun post edilmesi durumu kontrol edildi, post edildikten sonra form elemanları tek tek değişkenlere alındı. Bu değişkenlerin boş bırakılmaması için bir if bloğu ile içerikleri kontrol edildi. Ardından gerekli SQL komutu yazılıp veri tabanına gönderildi böylece verinin tabloya eklenmesi sağlandı. Bununla beraber konum ve sensör verilerinin tutulduğu tablodan verilerin çekilmesi ve mail olarak atılması için gereken değişkenler tanımlandı. Gerekli SQL komutu yazıldı ve veri tabanına gönderildi. Şekil 3.28’de bu kısma kadar olan bölümü görmekteyiz. Burada ekle sorgusunun bir cevap dönmesi beklenmediğinden bir değişkene kaydedilmedi fakat bul sorgusu tabloyu döndürdüğünden kayıt değişkeni ile saklandı.

```
if($_POST){
    $isim = $_POST["be_isim"];
    $sehir = $_POST["be_sehir"];
    $telefonNo = $_POST["be_tel"];
    $mail_input = $_POST["be_mail"];

    $link = "https://www.google.com/maps/search/?api=1&query=";
    $enlem;
    $boylam;
    $grup;
    $semt;

    $semtMesaj = "";
    $tSemtler = [];
    $kSemtler = [];
    $tsi = 0; $ksi = 0;

    if($isim == "" || $sehir == "" || $telefonNo == "" || $mail_input == ""){
        echo '<br>'. "Boş yer bırakılamaz.";
    }
    else{
        $ekle = "INSERT INTO `aboneler` (`ID`, `isim`, `sehir`, `tel_no`, `email`) VALUES (NULL, '$isim', '$sehir', '$telefonNo', '$mail_input')";

        $baglan->query($ekle);

        $bul = "SELECT * FROM konumsensor";
        $kayit = $baglan->query($bul);
    }
}
```

**Şekil 3.28.** SQL Komutları ve Değişkenler

Öncelikle bul sorgusundan dönen değer if bloğu ile kontrol edildi. Boş değer dönmez ise bul sorgusundan dönen tablo elemanlarını satır satır gezmek için fetch\_assoc() metodu kullanıldı. Bu metot ile her bir tablo satırına bir while döngüsü ile ulaşıldı ve mail için lazım olan enlem, boylam, grup, semt verileri değişkenlere atandı. Ardından grup değişkeni kontrol edilerek her bir satırdaki elemanın bölgesi anlaşıldı. Turuncu ve kırmızı alandaki bölgeler bir diziye kayıt edildi. Bu kayıt işlemi yapılırken HTML etiketi olarak a etiketi kullanıldı ve link, enlem, boylam, semt değişkenleri kullanılarak semt isminin link olarak kaydedilmesi sağlandı. Kırmızı bölge için ve turuncu bölge için ayrı iki dizi kullanıldı, böylece while döngüsünün sonunda turuncu bölgeleri

ve kırmızı bölgeleri bir HTML a etiketi olarak tutan iki dizi oluştı. Şekil 3.29’de bu döngüyü görmektesiniz.

```
if($kayit->num_rows>0){  
    while($satir = $kayit->fetch_assoc()){  
        $enlem = $satir["enlem"];  
        $boylam = $satir["boylam"];  
        $grup = $satir["grup"];  
        $semt = $satir["semt"];  
  
        if($grup == 1){  
        }  
        else if($grup == 2){  
            $tsemtler[$tsi] = '<a href="'. $link. "'>'. $enlem. ", ". $boylam. "'>'. $semt. '</a>';  
            $tsi++;  
        }  
        else if($grup == 3){  
            $ksemtler[$ksi] = '<a href="'. $link. "'>'. $enlem. ", ". $boylam. "'>'. $semt. '</a>';  
            $ksi++;  
        }  
    }  
}  
else{  
    echo "Veri Tabanı Boş.";  
}
```

**Şekil 3.29.** Konum ve Sensör Verilerinin Alınması

Bu aşamadan sonra geriye mail içeriğinin doldurulması ve gönderilmesi kaldı. İlk olarak mailin içeriğini dolduracak mesaj ayarlandı. Bu mesajda bir diziye kaydettiğimiz turuncu ve kırmızı bölgeler de gönderileceğinden, mesajın bölümleri ayrı değişkenlerde saklanıp en son birleştirildi. Bu dizilerdeki elemanlar for döngüsü ile kırmızı ve turuncu bölgeler ayrı olmak üzere tek bir değişkende birleştirildi. Böylece elde edilen beş farklı değişken en son toplanarak mail içeriği oluşturuldu. Daha sonra addAdres metodu ile mailin atılacağı adres belirlendi. Buraya mail adresi olarak formdan alınan veri gönderildi. Bölgeler için kullanılan HTML a etiketinin mail içeriğinde kullanılabilmesi için isHTML metoduna true değeri gönderildi. Ardından mailin başlığı ve içeriği belirtilerek send metodu aracılığıyla mailin gönderilmesi sağlandı. Mail gönderilirken sorun çıkmazsa echo komutu ile ‘mail gönderildi’ yazısı web sayfasında gösterildi, aksi halde ‘maalesef olmadı’ yazısı gösterildi. Şekil 3.30 ve şekil 3.31’de bu işlemlerin yapıldığı kodları görmektesiniz.



```

$baslangic = "<h1>Hoşgeldiniz.</h1>
<p> Platforma hoşgeldiniz, sağlığınız için elimizden geleni yapıyor, yüksek doğrulukla verileri sizlerle paylaşıyoruz.Marmaris'de astım
hastaları için dikkatli olunması gereken ve tehlikeli bölgeler aşağıdaki gibidir: </p>
<p> Dikkatli Olunması Gereken Bölgeler:</p>";
$tehlikeliString = "<p> Tehlikeli Bölgeler:</p>";
$son = "<p> Ayrıca istediğiniz zaman Bilgi yazıp 0544 578 3119 no'lu numaraya SMS atarak, SMS ile bilgilendirilebilirsiniz.</p>";

$tSemtSonString = "";

for($i = 0; $i < $tsi; $i++){
    if($tsi - $i == 1){
        $tSemtSonString .= $tSemtler[$i];
    }
    else{
        $tSemtSonString .= $tSemtler[$i].", ";
    }
}

$kSemtSonString = "";

for($i = 0; $i < $ksi; $i++){
    if($ksi - $i == 1){
        $kSemtSonString .= $kSemtler[$i];
    }
    else{
        $kSemtSonString .= $kSemtler[$i].", ";
    }
}

$mesaj = $baslangic."".$tSemtSonString."".$tehlikeliString."".$kSemtSonString.$son;

```

**Şekil 3.30.** Mail İçeriğinin Oluşturulması

```

$mail->addAddress(address:$mail_input, name:$isim);
$mail->isHTML(true);
$mail->Subject = "Tehlikeli Bolgeler";
$mail->Body = $mesaj;

if($mail->send()){
    echo '<div class="mesaj">Mail gönderildi.</div>';
}
else{
    echo '<div class="mesaj">Malesef olmadı.</div>';
}
}
?>

```

**Şekil 3.31.** Mailin Gönderilmesi

Böylece projenin web kısmı da tamamlanmış oldu. Bu bölümde amaçlanan statik verilerle bir platform kurmaktı ki bu başarıldı. Lakin projenin gerçek hayatta bir anlam kazanması için başlık 3.7 ve 3.8’de de belirtildiği gibi veriler dinamik bir şekilde alınmalı ve işlenmeli. Burada PHP kısmında ise konum ve sönör verileri tabloya statik olarak eklense bile mail gönderilirken dinamik bir yapı kuruldu. Veri tabanından verilerin çekilmesi ve anlık olarak çekilen bu verilere göre mail içeriğinin değişmesi sağlandı. Kısaca projenin donanım desteği olduğu takdirde işe yarar bir platform olacağı gösterilmiş oldu.

#### 4. SONUÇLAR

Yapılan bu çalışmada gündelik hayatta yaşadığımız sağlık sıkıntılarını teknolojiyi kullanarak en az seviyeye getirmek amaçlanmıştır. Projede astım hastaları hedef alınarak, ilk bu hastaların olumsuz etkilendiği gazlar göz önüne alındı ve gerekli gaz modülleri temin edildi. Belirli bölgelerden gaz sensörleri yardımıyla hava kirliliği verileri SD kart modülü ile kayıt edildi. Ardından kullanıcılara hava kirliliğinin kötü olduğu bölgelerinin risk bildirimini yapmak için haberleşme modülü olan SIM808 kullanıldı. Böylece kullanıcı bilgi almak istediğinde belirlenen telefon numarasına mesaj atarak riskli bölgeler hakkında bilgi alabilecek. Veriler alındıktan sonra ayrıca bunları kullanıcıya göstermek için bir web platformu oluşturuldu. Burada kullanıcı Google Map ile anlık olarak harita üzerinde riskli bölgeleri görebilecektir. Aynı zamanda kullanıcı abonelik formunu doldurduğu takdirde mail adresine riskli ve uzak durulması gereken bölgelerin konumları gönderilecektir.

Bütün aşamalar adım adım gerçekleşmiştir ve her adım sonunda gerekli kontroller yapılmıştır. Proje her ne kadar astım hastalarını hedef alsada olumsuz gazlardan etkilenebilecek birçok akciğer hastalıkları içinde geliştirilebilir. Projenin tamamlanmasının ardından elde edilen sonuca göre astım hastaları için yapılan bu proje direkt bir çözüm olmasada kişide bu hastalığın oluşmasının veya ilerlemesinin önüne geçebilir.

## KAYNAKLAR

- [1] İrkil S., ASTIMLI HASTALARDA EĞİTİMİN YAŞAM KALİTESİNE ETKİSİ, Kocaeli Üniversitesi Sağlık Bilimleri Enstitüsü, 15-18, 2006
- [2] Mermit Çilingir B., HAVA KİRLİLİĞİ VE AKCİĞER, Çağdaş Tıp Dergisi, 6(1-Ek (Olgu Sunumları)): 131-137, 2016
- [3] (<https://www.artistanbul.io/blog/2016/11/22/egitimde-arduinonun-yakin-tarihcesi/>)
- [4] (<https://simcom.ee/modules/gsm-gprs-gnss/sim808/>)
- [5] (<https://tr.wikipedia.org/wiki/HTML>)
- [6] (<https://tr.wikipedia.org/wiki/CSS>)
- [7] (<https://www.hostinger.web.tr/rehberler/javascript-nedir/>)
- [8] (<https://www.hostinger.web.tr/rehberler/php-nedir/>)
- [9] ([https://tr.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://tr.wikipedia.org/wiki/Visual_Studio_Code))
- [10] (<https://www.hosting.com.tr/blog/xampp/>)
- [11] (<https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/arduino-201/softwareserial-kutuphanesiyle-haberlesme>)
- [12] (<https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/arduino-401/spi-protokolu>)
- [13] (<https://www.robocombo.com/blog/icerik/arduino-ile-micro-sd-kart-modulu-kullanimi>)
- [14] (<https://www.simcom.com/product/SIM808.html>)
- [15] (<https://developers.google.com/maps/documentation/javascript/examples/circle-simple>)
- [16] (<https://muhammedyaman.com/phpmailer-kullanimi-php-smtp-mail-gonderme/>)

## ÖZGEÇMİŞ

**Ekrem BALI**



### Kişisel Bilgiler

**Okul Numarası:** 175260071

**Doğum Tarihi:** 11.05.1999

**Doğum Yeri:** OSMANGAZİ / BURSA

### İletişim Bilgileri

**Telefon:** 0532 012 29 37

**E-posta:** ekrem.bali16@outlook.com

**Adres:** Çiftehavuzlar Mahallesi 1.Ocak Sokak No:15 Kat:3 Osmangazi/BURSA

### Eğitim Durumu

**Lisans Öğrenimi:** Fırat Üniversitesi - (İkinci Öğretim)  
Mühendislik Fakültesi-Bilgisayar Mühendisliği

**Yabancı Diller:** İngilizce (Orta)

### İş Deneyimi

#### 2021 Zorunlu Staj-1

4B Yazılım Bilgi Teknolojileri | Van Bilişim

## **DOĞUKAN AYDOĞDU**



### **Kişisel Bilgiler**

**Okul Numarası:** 175260061

**Doğum Tarihi:** 18.08.1999

**Doğum Yeri:** EFELER / AYDIN

### **İletişim Bilgileri**

**Telefon:** 0544 578 31 19

**E-posta:** doguaa09@gmail.com

**Adres:** Mesudiye Mahallesi 1660 Sokak No:16 Efeler / Aydın

### **Eğitim Durumu**

**Lisans Öğrenimi:** Fırat Üniversitesi - (İkinci Öğretim)  
Mühendislik Fakültesi-Bilgisayar Mühendisliği

**Yabancı Diller:** İngilizce (Orta Seviye)

### **İş Deneyimi**

#### **2020 Zorunlu Staj-1**

MAVA Yazılım Mobil Uygulama Çözümleri