



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پروژه درس آزمایشگاه سخت افزار

عنوان:

مجموعه مستندات سیستم اندازه گیری علائم حیاتی بدن

نگارندهان:

علیرضا تاج میر ریاحی، امیر مهدی نامجو، صبا هاشمی

استاد گرامی:

جناب آقای دکتر اجلالی - جناب آقای دکتر فصحتی

زمستان ۱۴۰۰ و بهار ۱۴۰۱

اللهُ أَكْبَرُ

فهرست مطالب

۶	۱ مقدمه
۷	۲ معناری سیستم
۹	۱-۲ طراحی و پیاده‌سازی سخت‌افزار
۹	۱-۱-۲ سنسورهای محیطی
۱۱	۱-۲-۲ سنسورهای بدن
۱۷	۳-۱-۲ بسته بندی
۱۹	۲-۲ طراحی و پیاده‌سازی سرور
۲۱	۲-۳ طراحی و پیاده‌سازی اپلیکیشن
۲۲	۳ قیمت
۲۴	۴ جمع‌بندی

فهرست شکل‌ها

۱-۲	معماری سطح بالای سیستم	۸
۲-۲	اتصال سنسور DHT11	۱۰
۳-۲	اتصال سنسور MQ135	۱۰
۴-۲	اتصال سنسور Max30205	۱۲
۵-۲	اتصال سنسور Max30102	۱۴
۶-۲	نحوه اتصال Lead های سنسور نوار قلب به بدن	۱۵
۷-۲	اتصال سنسورهای بدن به رزبری‌پای و آردوینو در کنار هم	۱۵
۸-۲	فشارسنج	۱۷
۹-۲	بسته‌بندی طراحی شده در نرم‌افزار Tinkercad	۱۸
۱۰-۲	محصول به همراه تمامی سنسورها	۱۸
۱۱-۲	مستندات Swagger	۲۰

فهرست جداول

۱-۳ جدول قیمت محصول (قیمت‌ها به واحد هزارتومان) ۲۳

فصل ۱

مقدمه

هدف از این پروژه، ارائه محصولی جامع برای اندازهگیری خودکار و نظارت بر علائم حیاتی بیمار به همراه شرایط محیطی است. این پروژه به شکل کلی‌تر، قابلیت استقرار در محیط‌های بیمارستانی و همچنین محیط‌های خانگی برای افرادی که نیازمند مراقبت ویژه هستند را دارد.

محصول توانایی اندازهگیری دما، رطوبت و آلودگی هوا را به عنوان عوامل محیطی اثرگذار در شرایط بیمار دارد. در کنار آن، دمای بدن، ضربان قلب، اکسیژن خون و نوار قلب به طور خودکار ثبت شده و امکان وارد کردن فشار خون که باید به صورت دستی و توسط متخصص اندازه گیری بشود، نیز وجود دارد.

همه این داده‌ها، به تفکیک بیمار در دیتاییسی تجمعی شده و از طریق نرم‌افزار موبایلی طراحی شده که قابلیت استفاده به صورت وب‌اپلیکیشن را هم دارد، امکان مشاهده آنی این اطلاعات برای پزشک یا سایر کادر درمانی میسر خواهد بود.

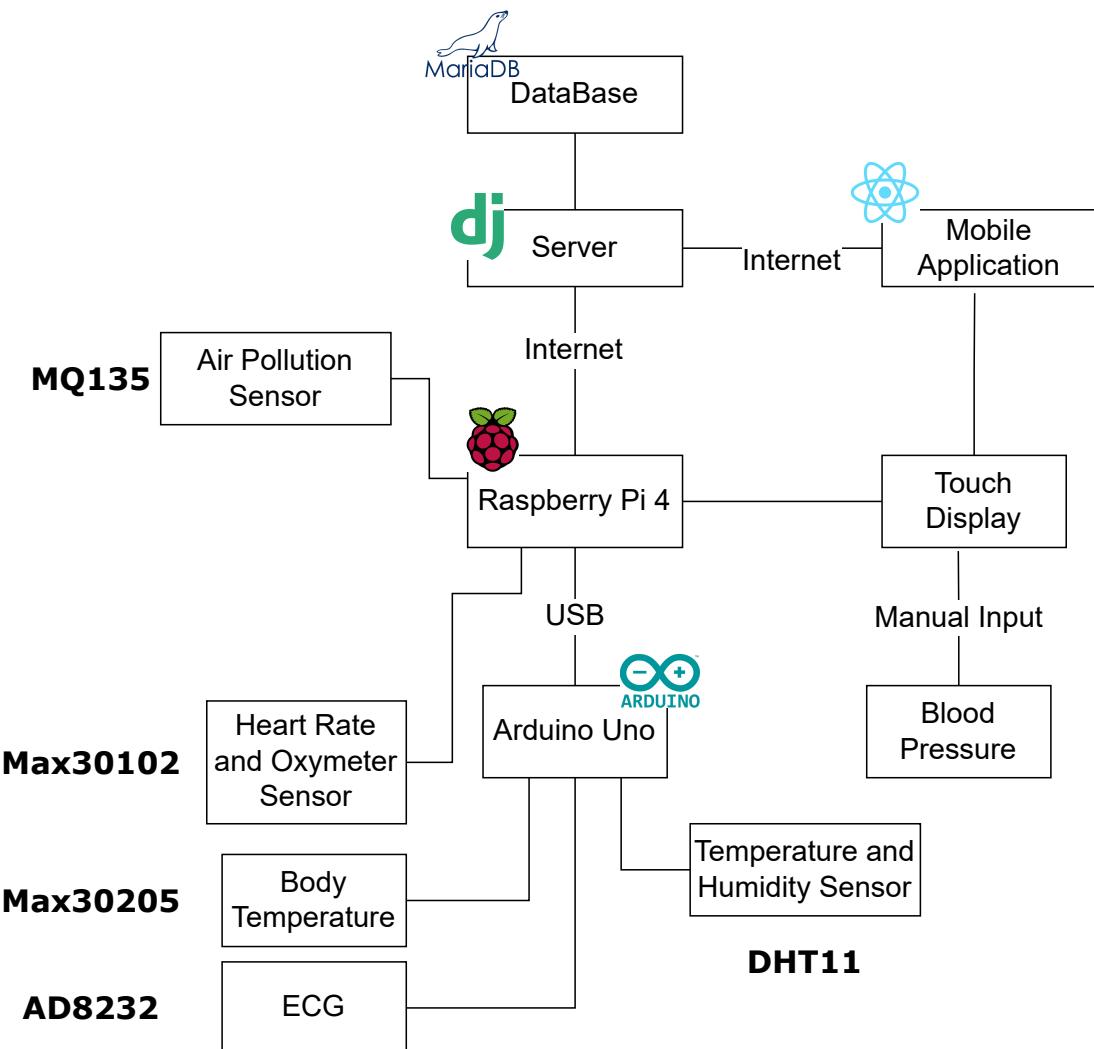
مزیت رقابتی اصلی این محصول، تجمعی سیستم‌های جمع‌آوری داده‌ها در یک محصول به همراه ارائه قابلیت نظارت همزمان و گزارش‌گیری از طریق اپلیکیشن طراحی شده است. نمونه‌های موجود در بازار، هیچ‌کدام به صورت یک محصول جامع شامل همه سنسورها نیستند و به علاوه اکثر آنان قابلیت ارتباط برقرار کردن با نرم‌افزارهای موبایلی که به راحتی قابل استفاده باشند را ندارند. تجمعی سیستم‌های جمع‌آوری داده به همراه نرم‌افزار کاربرپسند که به راحتی قابل استفاده باشد، مزایای رقابتی اصلی این محصول هستند.

فصل ۲

معماری سیستم

سیستم طراحی شده توسط ما از سه قسمت اساسی تشکیل شده است. قسمت سخت‌افزاری متشکل از رزبری‌پای، آردوینو و سنسورهای مختلف که برای اندازه‌گیری علائم حیاتی و شرایط محیطی استفاده می‌شوند. سرور که برای دریافت داده‌ها و ساماندهی آنان استفاده شده است و همچنین اپلیکیشن موبایلی که برای نمایش داده‌های جمع‌آوری شده و نظارت همزمان بر آن‌ها استفاده شده است.

معماری سطح بالای سیستم در شکل ۱-۲ قابل مشاهده است.



شكل ۲ - ۱ : معماری سطح بالای سیستم

۱-۲ طراحی و پیاده‌سازی سخت‌افزار

اصلی‌ترین قسمت این پروژه، طراحی و پیاده‌سازی قسمت‌های سخت‌افزاری آن است. در زیر لیستی از قطعات سخت‌افزاری مورد استفاده آمده است و پس از آن توضیحاتی در مورد هر یک از سنسورها و نحوه کارکرد و راهاندازی آن ذکر شده است.

- برد Raspberry Pi 4

- برد Arduino UNO

- صفحه نمایش لمسی 7 اینچ مخصوص Raspberry Pi

- سنسور آلودگی هوا MQ135

- سنسور دما و رطوبت هوا DHT11

- سنسور ضربان قلب و اکسیژن خون Max30102

- سنسور دمای بدن Max30205

- سنسور نوار قلب AD8232

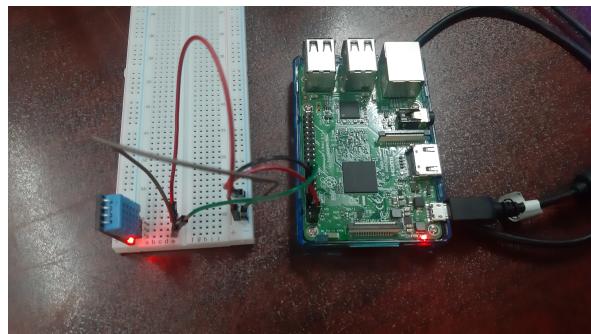
- فشار سنج و گوشی پزشکی

۱-۱ سنسورهای محیطی

د، سنسور محیطی اصلی در این پروژه وجود دارند. سنسور MQ135 که وظیفه اندازه‌گیری آلودگی هوا را داشته و سنسور DHT11 که وظیفه اندازه‌گیری دما و رطوبت را دارد. سنسور آلودگی هوا به آردوینو متصل شده و سنسور اندازه‌گیری دما و رطوبت هوا مستقیماً به رزبری‌پای متصل می‌شود.

سنسور دما و رطوبت هوا

سنسور مورد استفاده برای این بخش، DHT11 است که از قابلیت انتقال داده به صورت دیجیتال پشتیبانی کرده و برای همین به راحتی مطابق شکل ۲-۲ به رزبری‌پای متصل می‌شود.



شکل ۲-۲: اتصال سنسور DHT11

برای خواندن مقادیر از کتابخانه‌ی Adafruit-Blinka^۱ استفاده شده است. این کتابخانه با مشخص کردن پین متصل به سنسور، به راحتی امکان خواندن دما و رطوبت هوا را به ما می‌دهد. کد اصلی مربوط به این قسمت در زیر آورده شده است:

```
dht11_sensor = adafruit_dht.DHT11(board.D23)
temp = dht11_sensor.temperature
humidity = dht11_sensor.humidity
```

سنسور آلودگی‌هوا

با توجه به این که سنسور MQ135 خروجی اصلی خود را به صورت آنالوگ تحویل داده و حتی رابط I2C هم ندارد، آن را به برد آردوینو متصل کرده و از طریق اتصال رزبری‌پایی به آردوینو با پورت USB کد مربوط به آن را از طریق رزبری به برد آردوینو انتقال داده و داده‌های لازم را دریافت می‌کنیم.

نحوه اتصال این سنسور در کنار سنسور قبلی در شکل ۲-۳ قرار دارد.



شکل ۲-۳: اتصال سنسور MQ135

کد اصلی مربوط به این قسمت در زیر آمده است:

<https://pypi.org/project/Adafruit-Blinka/>

در کد آردوینو، مقادیر مربوط به این سنسور هر ۵ ثانیه خوانده می‌شود:

```
#include <Wire.h>

void setup() {
    Serial.begin(9600);
    Wire.begin();
}

int counter = 0; // 1 milisecond

void loop() {
    if (counter % 5000 == 0) // 5 second
    {
        int pollution = analogRead(A0);
        Serial.print("pollution , ");
        Serial.println(pollution);
    }
    counter += 10;
    delay(10);
}
```

سپس در کد پایتون روی رزبری، این مقادیر روی یک فایل ریخته می‌شود و پس از آن به همراه مقادیر باقی سنسورها به سرور ارسال می‌گردد.

۲-۱-۲ سنسورهای بدن

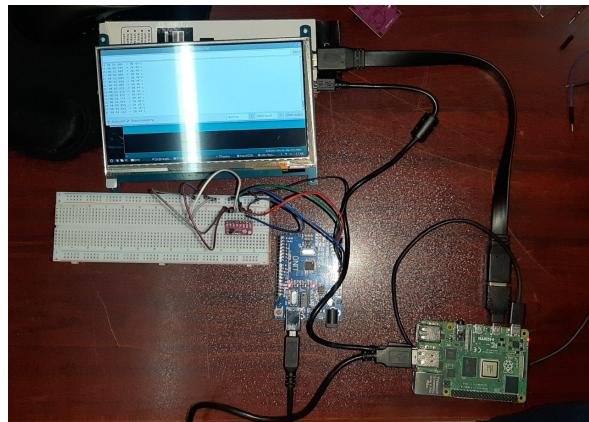
سه سنسور اصلی برای علائم مربوط به بدن انسان در این پروژه وجود دارند. سنسور دمای بدن Max30205، سنسور ضربان قلب و اکسیژن خون Max30102 و سنسور نوار قلب یعنی AD8232. سنسور Max30102 مستقیماً به رزبری‌پای متصل می‌شود ولی دو سنسور دیگر از طریق آردوینو با رزبری‌پای ارتباط برقرار

می‌کنند.

سنسور دمای بدن

برای سنجش دمای بدن از سنسور Max30205 استفاده شده است. با تماس انگشت به آن، بعد از مدتی دمای سنسور با دمای انگشت همدما شده و دمای بدن را نشان خواهد داد. در صورت عدم تماس هم می‌توان از آن برای مشاهده دمای محیط استفاده کرد.

این سنسور به آردوینو متصل شده و از طریق اتصال آردوینو به رزبری‌پای، اطلاعات آن را مشاهده می‌کنیم. در شکل ۲-۴ تصویر اتصال این سنسور قرار دارد.



شکل ۲-۴: اتصال سنسور Max30205

برای خواندن مقادیر این سنسور از کتابخانه Protocentral_MAX30205^۴ استفاده شده که امکان خواندن دمای بدن را از طریق آردوینو می‌دهد.

```
#include <Wire.h>
#include "Protocentral_MAX30205.h"
MAX30205 tempSensor;

void setup() {
    Serial.begin(9600);
    Wire.begin();
}
```

^۴ https://github.com/Protocentral/Protocentral_MAX30205

```

while (! tempSensor . scanAvailableSensors ()) {
    Serial . println ("Couldn't find the sensor . " );
    delay (3000);
}

int counter = 0; // 1 milisecond

void loop () {
    if (counter % 5000 == 0) // 5 second
    {
        float temp = tempSensor . getTemperature ();
        Serial . print ("temp , " );
        Serial . println (temp , 2);
    }
    counter += 10;
    delay (10);
}

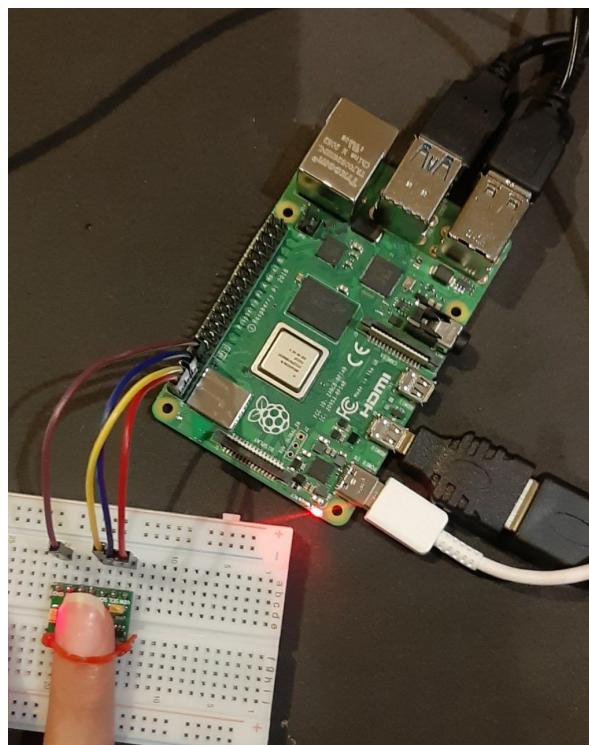
```

سنسور ضربان قلب و اکسیژن خون

برای این بخش از سنسور Max30102 استفاده شده است. این سنسور با کمک دو چراغ کوچک قرمز و مادون قرمز، ضربان قلب و درصد اشباع اکسیژن در خون (SpO₂) را اندازه‌گیری می‌کند. این سنسور بدون مشکل از طریق I2C به رزبری‌پای متصل می‌شود. نحوه اتصال این سنسور را در شکل ۵-۲ مشاهده می‌کنید.

برای راهاندازی این سنسور از کدهای مخزن متن باز doug-burrell/max30102^۳ با اندکی تغییرات استفاده شده است. این مخزن با خواندن مقادیر سنسورهای قرمز/مادون قرمز و پردازش آنها، مقادیر ضربان قلب و اکسیژن خون را به طور دقیق محاسبه می‌کند. کدهای مربوط به این مخزن در فایل‌های

^۳ <https://github.com/doug-burrell/max30102>



شکل ۲-۵: اتصال سنسور ۳۰۱۰۲ Max

امکانات این کتابخانه به hrcalc.py و max30102.py قرار دارند. از این کتابخانات این کد زیر در کد رزبری استفاده شده است:

```
from heartrate_monitor import HeartRateMonitor
```

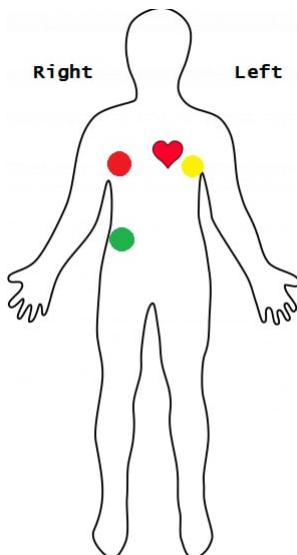
```
hrm = HeartRateMonitor(print_raw=False, print_result=False)  
hrm.start_sensor()
```

```
bpm = hrm.bpm
```

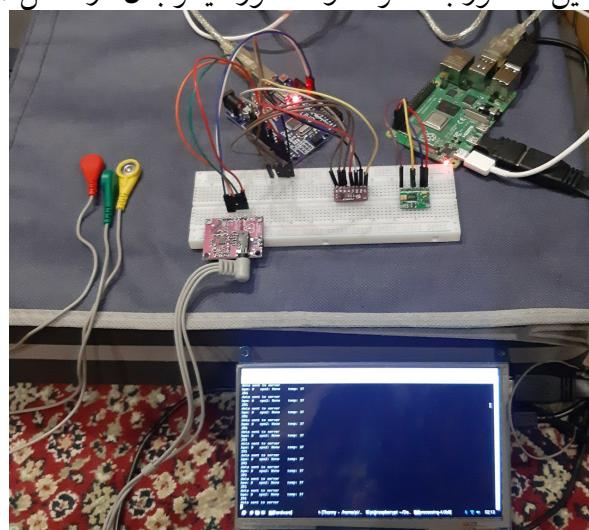
```
spo2 = hrm.spo2
```

سنسور نوار قلب

از سنسور AD8232 برای بدست آوردن ECG (نوار قلب) استفاده می‌شود. همراه این سنسور بسته Lead سه تایی اتصال به بدن وجود دارد. به این Lead ها باید پدهای مخصوص متصل شده و به بدن متصل شوند. برای استفاده از سنسور، آن را به آردوینو متصل می‌کنیم. Lead های آن هم باید مطابق شکل ۲-۶ به بدن بیمار متصل بشوند.



شکل ۲-۶: نحوه اتصال Lead های سنسور نوار قلب به بدن
همچنین نحوه اتصال این سنسور به همراه دو سنسور دیگر بدن در شکل ۲-۷ قرار دارد.



شکل ۲-۷: اتصال سنسورهای بدن به رزبریپای و آردوینو در کنار هم
کدهای آردوینو مربوط به این سنسور در زیر آمده است. هر ۱۰ میلی ثانیه مقادیر مربوط به این

سنسور از طریق آردوینو خوانده می‌شود.

```
#include <Wire.h>
void setup() {
    Serial.begin(9600);
    Wire.begin();
    pinMode(10, INPUT);
    pinMode(11, OUTPUT);
}
int counter = 0; // 1 milisecond
void loop() {
    if (counter % 10 == 0) // 10 milisecond
    {
        Serial.print("ecg,");
        if(digitalRead(10) == 1 || digitalRead(11) == 1) {
            Serial.println('!');
        }
        else {
            int ecg = analogRead(A1);
            Serial.println(ecg);
        }
    }
    counter += 10;
    delay(10);
}
```

اندازهگیری فشار خون

با توجه به این که اندازهگیری فشار خون به شکل دقیق معمولاً توسط فرد آموختن دیده انجام می‌گیرد و ساخت فشارسنج اتوماتیک از حوزه کاری این محصول خارج بوده و خود یک محصول جداگانه است، فشار در این محصول به شکل انسانی با استفاده از فشارسنج اندازهگیری شده و در منویی که برای ثبت فشار در نرمافزار قرار گرفته است، وارد می‌شود. تصویر فشارسنج در شکل ۲-۸ قرار گرفته است.

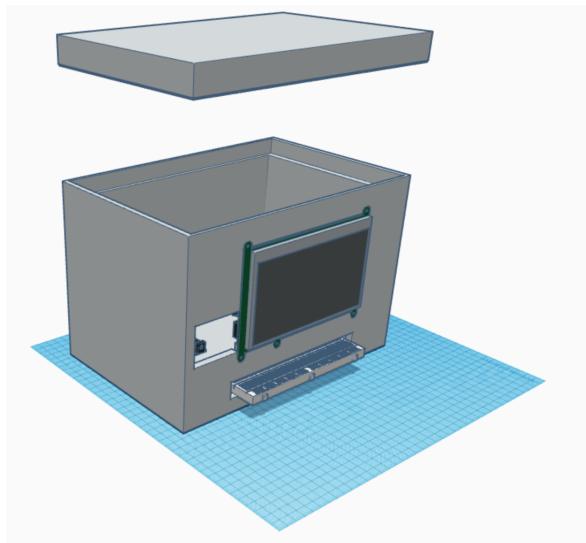


شکل ۲-۸: فشارسنج

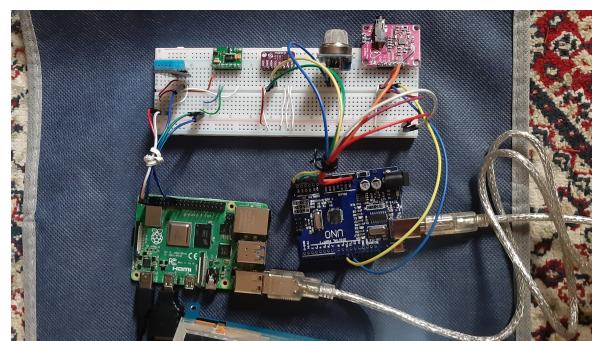
۳-۱-۲ بسته بندی

برای بسته‌بندی، با توجه به هماهنگی‌های صورت گرفته قرار است که از پرینتر سه‌بعدی استفاده شود. از این رو قسمت بسته‌بندی محصول هنوز کامل نیست. در شکل ۹-۲ طرح آماده شده برای بسته‌بندی را مشاهده می‌کنید.

وضعیت فعلی محصول به همراه تمامی سنسورها در شکل ۱۰-۲ آمده است.



شکل ۲-۹: بسته‌بندی طراحی شده در نرم‌افزار Tinkercad



شکل ۲-۱۰: محصول به همراه تمامی سنسورها

۲-۲ طراحی و پیاده‌سازی سرور

سرور در این پروژه وظیفه دریافت اطلاعات از رزبری‌پای، ذخیره آن‌ها در پایگاه داده و تحويل دادن آن‌ها با فیلترها و به شکل مناسب به نرم‌افزار طراحی شده را دارد.

برای طراحی سرور از فریم‌ورک Django Rest Framework و Django استفاده شده است که به ما امکان طراحی سریع و در عین حال اصولی سروری که برای این پروژه مناسب باشد را می‌داد. برای پایگاه‌داده هم از MariaDB استفاده شده است که براساس MySQL توسعه یافته است و با قواعد MySQL هماهنگی کامل دارد. برای راحتی کار توسعه سرور و نصب نیازمندی‌های آن، از Docker Compose استفاده شده است تا به راحتی همه اجزای مختلف سرور مستقل از سیستمی که توسعه روی آن صورت می‌گیرد به شکل مناسب و سریع استقرار یابد.

همچنین برای این که نیاز به استفاده از دستورات Docker کمینه شود، یک فایل Makefile هم نوشته شده است که با دو دستور اصلی make build و make up، راهاندازی اولیه سرور و بالا آوردن آن بعد از راهاندازی اولیه قابل انجام است. همچنین با دستور make down می‌توان سرور را خاموش کرد.

سرور پروژه از دو قسمت اصلی تشکیل شده است. بخش Patients که در آن مدل مربوط به بیماران (شامل نام و نام خانوادگی، شماره تلفن، قد، وزن و جنسیت) در فایل models.py در فایل تعريف شده است. همچنین با استفاده از کلاس‌های Django Rest Framework تمامی کارهای مربوط به ایجاد، آپدیت، حذف و دریافت اطلاعات هویتی به بیماران در این قسمت و فایل views.py انجام می‌شود.

بخش دیگر Records است. در این جا مدل اصلی برای ذخیره داده‌های بیماران و همچنین داده‌های محیطی تعريف شده است. همچنین در قسمت views.py علاوه بر تعريف دو کلاس براساس Django Framework Rest که امکان عملیات‌های یجاد، آپدیت، حذف و دریافت را برای داده‌های بیماران فراهم کنند، سه تابع برای دریافت آخرین داده، دریافت آخرین داده‌ها از زمان مشخص شده و دریافت داده‌ها با تعیین فیلترهای مختلف (زمان، بیمار و سنسورهای خاص) تعريف شده است. همچنین تابعی که آخرین داده را برای ما فراهم می‌کند، برای هر فیلد مستقل به دنبال آخرین داده می‌گردد تا اگر در داده‌هایی که برای آخرین زمان ثبت شده‌اند، برای برخی از سنسورها داده‌ای وجود نداشت، آخرین داده موجود گزارش شود. در صورتی که هیچ داده‌ای وجود نداشته باشد، مقادیر پیش‌فرض ثبت شده برای آن فیلد گزارش می‌شوند.

علاوه بر این، با استفاده از ابزار Swagger مستندات خودکار برای استفاده از API‌ها هم ایجاد شده

است. در صورتی که سرور به صورت محلی اجرا شود، این مستندات از آدرس `localhost:8000/swagger` در دسترس خواهد بود. تصویر آن را در شکل ۱۱-۲ مشاهده می‌کنید.

شکل ۱۱-۲ : مستندات Swagger

۳-۲ طراحی و پیاده‌سازی اپلیکیشن

فصل ۳

قیمت

یکی از مسائل مهم در طراحی محصول قیمت آن است. البته با توجه به این که این محصول به صورت نمونه اولیه طراحی شده است، طبیعتاً قیمت تمام شده آن از محصولی که بخواهد تولید عمده بشود بالاتر خواهد بود. در جدول ۱-۳ قیمتی تخمین زده شده و هزینه نهایی پروژه آورده شده است.

عمده تفاوت قیمت بین اعداد تخمینی و عدد نهایی، به دلیل تغییر رزبری‌پای از ۳B به ۴، اضافه شدن آردوبینو و همچنین اضافه شدن فشارسنج دستی صورت گرفته است.

ردیف	قطعه	قیمت تخمینی	قیمت نهایی
۱	MQ-135	۳.۳۸	۳.۳۸
۲	MPS20N004D	۵.۲۶	.
۳	DHT11 or KY-015	۴۰	۴۰
۴	MAX30205	۹.۲۹۰	۹.۲۹۰
۵	AD8232	۱۲۸	۱۲۸
۶	ECG Electrode	۹۶	۹۶
۷	MAX30102	۳.۱۶۳	۳.۱۶۳
۸	Raspberry Pi LCD	۱۴۸۰	۲۰۰۰
۹	Raspberry Pi 3B	۳۰۰۰	.
۱۰	Raspberry Pi 4	.	۴۰۰
۱۱	Arduino Uno	.	۴۰۰
۱۲	Breadboard	۰.۴۷	۰.۴۷
۱۳	Wires	۱۰۰	۱۰۰
۱۴	Resistors	۷	۱۰
۱۵	ADS1115	۱۰۴	.
۱۶	Sphygmomanometer	.	۷۰۰
۱۷	ECG Pads	۱۲	۵۰
۱۸	Total	۰.۵۵۸۸	۸۰۶۹

جدول ۳-۱: جدول قیمت محصول (قیمت‌ها به واحد هزار تومان)

فصل ۴

جمع‌بندی