

ابتدا به بررسی ۱-نتایج پروژه سپس ۲-نکات مهم در روند پروژه را بررسی میکنیم و ۳-قطعه کد هارو توضیح میدیم.

۱-نتایج پروژه:

جهت سهولت بررسی گزارش پروژه متن زیر به بخش مربوطه لینک گذاری شده است (البته به معنای این نیست که فقط در همون بخش از گزارش کار پروژه این مورد ها بحث شده طبیعتاً یک مورد ممکنه چندین جا بحث و بررسی شده باشه) سعی شده تا جای ممکن لینک این متن ها به جای درست در گزارش کار باشه فقط کافیه رو متن پایین کلیک کنید (هر ستاره یک لینک و معنی تیک هم یعنی این مورد در پروژه اجرا شده)

* جکیده و * مقدمه

A) Core Requirements (up to 70 points) • [BASE-OS] Install OS✓, secure user*, static IP*, SSH with keys*, disable root* (10) • [BASE-LAMP] Install and configure Apache/PHP/MySQL, limited DB user* (10) • [WEB-DASH] Implement a web dashboard* that shows system health* and event list*, and provide basic controls* to modify* server configuration through* the web (10 + 5) • [DB-SEED] Sample data and successful connection (5) ✓ • [DET-AUDIO/IMG] Detection script* (audio* or image) with DB insertion* (15) • [SVC-SYSTEMD] systemd* service with auto*-restart* and resource limits* (10) • [MQTT-PIPE] Publish resource status* and subscribe real-time display* + LWT test* (5) • [MQTT-TLS] Mosquitto* with TLS* and username/password* (5)

B) Quality* & Security* (up to 20 points) • [SEC-HARDEN] Documented & applied: basic UFW*/iptables*, service restriction*, minimal DB access* (5) • [ROBUST] Robustness*: error handling*, meaningful logs*, no crashes on network disconnects✓ (5) • [UX-CLEAN] Clean UI*, simple navigation✓, mobile*/desktop* readability (10)

C) Advanced / Extra Features (bonus up to 15 points) • [API-REST] Clean API✓ with simple authentication* and light rate-limiting* (5) • [DB-INDEX] Indexing* and simple benchmark* for query performance (3) • [WEB-AJAX] Live dashboard✓ auto-refresh✓ with AJAX/Fetch✓ (4) • [CNT-DOCKER] Docker container for detection script* or services* (4) • [ML-ADV] Advanced* ML model* beyond threshold*/frame* differencing (5)

لینک ویدیو ضبط شده با کیفیت اصلی :

<https://drive.google.com/file/d/1ZHTQbrVcPhzi7X64JAiOSaKOG1vU1GM/view?usp=sharing>

چکیده

این پروژه به طراحی و پیاده‌سازی یک سامانه‌ی لبه‌ای یکپارچه بر پایه‌ی **Orange Pi** می‌پردازد که همزمان نقش‌های وب‌سرور امن (**HTTPS/TLS**)، تشخیص رویداد صوتی و مانیتورینگ لحظه‌ای منابع را ایفا می‌کند. تمام مؤلفه‌های وب شامل **REST API**، اسکریپت‌های **JSON**، Apache/PHP برای برنامه‌ی وب و **Style/CSS** بر روی خود **Orange Pi** اجرا می‌شوند و از هر کلاینتی در شبکه به صورت امن در دسترس‌اند. تشخیص رویداد صوتی با اسکریپت Python متصل به میکروفون USB انجام می‌شود و امکان فعال/غیرفعال‌سازی الگوریتم **ML**، تنظیم نرخ نمونه‌برداری و مسدودسازی کلاس‌های صوتی ناخواسته بدون درج در **DB** از طریق داشبورد فراهم است. نتایج تشخیص با فیلدهای استاندارد نوع، زمان‌بندی ISO 8601، سطح اطمینان و در MySQL ذخیره و با ایندکس‌گذاری برای جستجو و پرس‌وجوی سریع بهینه‌سازی شده‌اند.

برای مانیتورینگ بلاذرنگ، **publisher** **Orange Pi** با استفاده از **psutil**، وضعیت **CPU, RAM**، دما را هر 3 ثانیه به بروکر **Mosquitto** (روی لپ‌تاپ) ارسال می‌کند؛ **TLS** و نام کاربری/گذرواژه در **MQTT** پیاده شده و سناریوی **LWT** نیز آزموده شده است. سرویس‌های تشخیص و انتشار با **systemd** به صورت خودکار در بوت اجرا می‌شوند، محدودیت منابع دارند و در صورت خطا خودترمیم هستند. داشبورد وب‌شکیل و واکنش‌گرا، علاوه بر مشاهده‌ی رویدادها و جستجوی پیشرفته، کنترل‌های سیستمی خاموش/اری استارت‌نمایش زنده‌ی منابع، **MQTT GUI** تعبیه‌شده، مدیریت کاربران با نقش‌های **AJAX live-**, **Dockerization**، نمایش آخرین زمان ورود و خروج امن را ارائه می‌دهد. افزون بر الزامات، **refresh** و **API** با احراز هویت و **rate-limit** سبک نیز پیاده‌سازی شده تا سامانه‌ای ایمن، پایدار، مقیاس‌پذیر و قابل توسعه در بستر **Edge Computing/IoT** حاصل شود.

مقدمه

رشد شتابان IoT نیازمند سامانه‌هایی است که پردازش نزدیک‌به‌داده، پایش بلاذرنگ و ارتباط امن را در یک پکیج کم‌هزینه و قابل اتکا ترکیب کنند. در این پروژه، با تکیه بر توان **Orange Pi** به عنوان یک SBC کارآمد، یک معماری کمینه اما کامل طراحی شد که تمام لایه‌ی وب و کنترل را بر روی همان گره لبه‌ای متمرکز می‌کند و تنها یک گره هسته‌ای (بروکر روی لپ‌تاپ) برای پیام‌رسانی بلاذرنگ در نظر می‌گیرد. بدین ترتیب، معماری منطقی سیستم به صورت زیر است:

گره لبه (Orange Pi) میزبانی (LAMP (Apache/PHP/MySQL)، اجرای وب‌سایت امن با **HTTPS/TLS** و **REST API (JSON)**، اسکریپت تشخیص صوت **MQTT Publisher**، **Python** برای تلمتری منابع، داشبورد **AJAX live-refresh** و واکنش‌گرا با **MQTT GUI** و **Shutdown/Restart** تعبیه‌شده. سیستمی (Shutdown/Restart).

گره هسته (لپ تاپ دانشجو) میزبانی بروکر Mosquitto با TLS و احراز هویت، کلاینت Subscriber جهت مشاهده‌ی زنده‌ی پیام‌ها و آزمون LWT

کلاینت‌های وب (هر دستگاه در شبکه) دسترسی امن به داشبورد API روی Orange Pi، با امکان جست‌وجو و مدیریت و مشاهده‌ی لحظه‌ای.

مسئله‌ی اصلی، تجمیع سه قابلیت کلیدی (وب‌서ور، تشخیص، مانیتورینگ) بر روی یک گره لبه‌ای با امنیت و قابلیت اطمینان بالا بود. برای این منظور:

۱. امنیت پایه به صورت مستند اعمال شد: ایجاد کاربر امن، SSH با کلید، غیرفعال‌سازی root، UFW/iptables

MQTT با TLS، ارتباطات DB؛ و وب‌سرور با گواهی TLS برای HTTPS

۲. قابلیت اطمینان و بهره‌برداری با systemctl تمیین شد اجرای خودکار در بوت، RestartPolicy مناسب،

Docker برای یکپارچگی اجرا و حمل‌پذیری افزوده محدودیت CPU/Memory و لاغ‌برداری معنادار؛ همچنین RestartPolicy محدودیت CPU/Memory و لاغ‌برداری معنادار؛ همچنین Docker برای یکپارچگی اجرا و حمل‌پذیری افزوده شد.

۳. کیفیت تجربه‌ی کاربری با یک UI شکیل و واکنش‌گرا، ناوبری ساده، MQTT GUI درون داشبورد و کنترل‌های مدیریتی کامل ارتقا یافت.

۴. مدیریت داده و کارایی با طراحی اسکیم‌ای رویداد، ایندکس‌گذاری هدفمند، فیلتر جست‌وجوی پیشرفته، و اعمال سیاست‌هایی چون غیرفعال‌سازی کلاس‌های صوتی ناخواسته (حذف درج) و تنظیم‌پذیری نرخ نمونه‌برداری-ML-Mode محقق شد.

۵. قابلیت توسعه با REST API سبک، خروجی JSON استاندارد ISO 8601 و timestamp برای محافظت از سرویس فراهم گردید.

JSON REST API، MySQL، HTTPS، PHP، Apache منطق کلاینت (JS/CSS) و داشبورد مدیریتی در کنار آن، بروکر Mosquitto روی Orange Pi all in one در شبکه می‌تواند به‌طور امن به داشبورد API دسترسی یابد. آزمون‌های اتصال از جمله LWT را تسهیل کند؛ هر کلاینت وب در شبکه می‌تواند به‌طور امن به داشبورد API دسترسی یابد.

در لایه‌ی تشخیص، مسیر صوت با Python و sounddevice پیاده‌سازی شده و با افزودن کنترل‌های عملیاتی فعال/غیرفعال‌سازی ML-mode، تنظیم sampling و بن‌کردن کلاس‌های ناخواسته از طریق UI، انعطاف بهره‌برداری افزایش یافته است. رویدادها با فیلدهای استاندارد در MySQL ثبت و با ایندکس‌گذاری روی ستون‌های پر تکرار (نظیر زمان/نوع رویداد) امکان جست‌وجو و پالایش سریع فراهم شده است. در لایه‌ی پایش منابع، اسکریپت Publisher روی Orange Pi

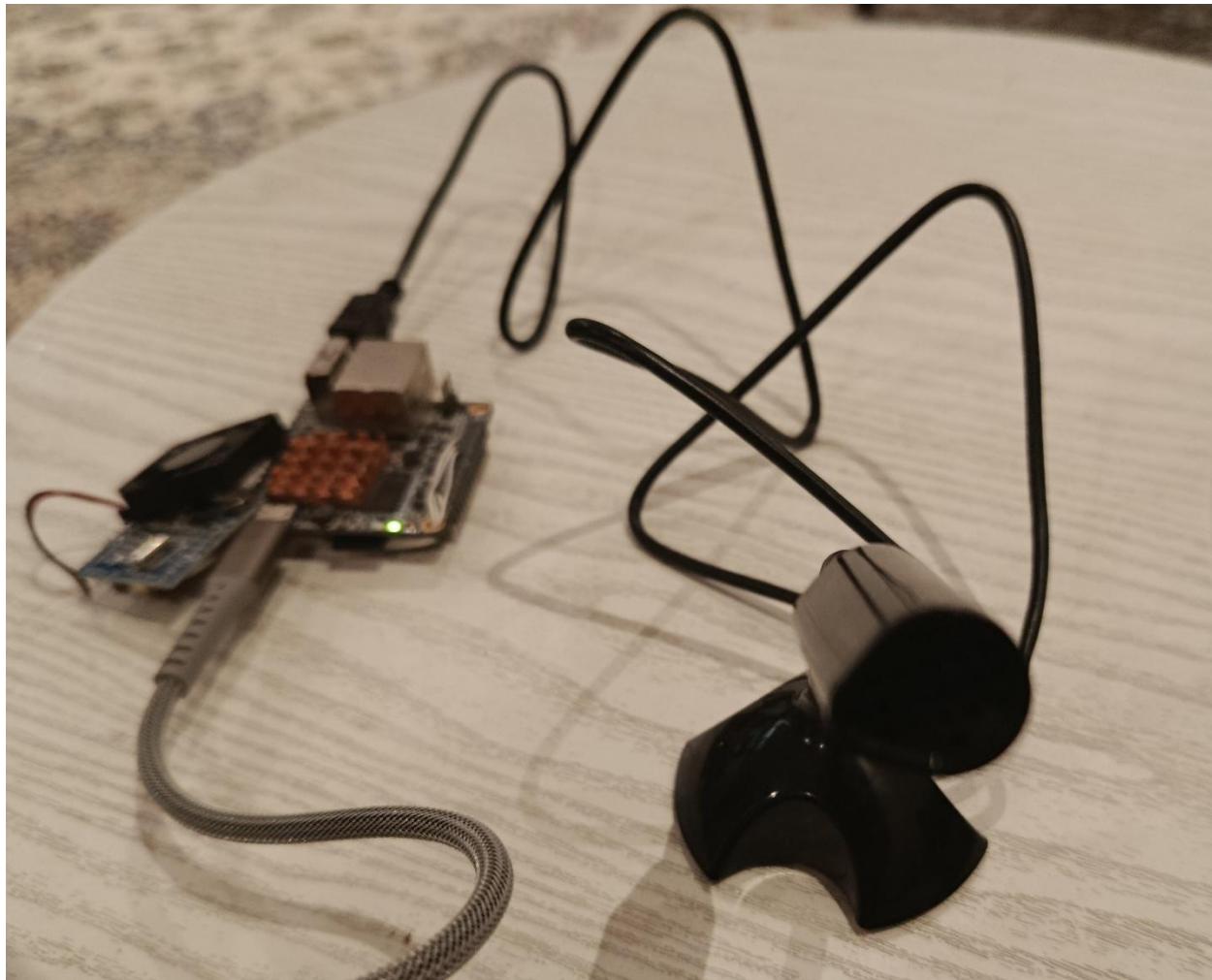
تکیه بر psutil ، تله‌متري CPU, RAM درون داشبورد دریافت‌ها را به صورت زنده نمایش می‌دهد.

در حوزه‌ی امنیت و پایداری، مجموعه‌ای از تدابیر اجرا شده است: کاربر امن، SSH با کلید و غیرفعال سازی root، حداقل‌سازی دسترسی DB، TLS برای MQTT و HTTPS برای وب. سرویس‌ها با systemctl دارای auto-start پس از بوت، سیاست‌های restart و محدودیت CPU/Memory هستند و لگ‌برداری معنادار دارند؛ Docker برای یکپارچگی اجرا و حمل‌پذیری به کار گرفته شده است. از منظر تجربه‌ی کاربری، یک UI تمیز و واکنش‌گرا با ناوبری ساده، به روزرسانی زنده (AJAX)، مدیریت کاربران با نقش‌های مجزا (RBAC)، نمایش آخرین زمان ورود و کنترل‌های سیستمی (Shutdown/Restart) ارائه شده است.

این کار نشان می‌دهد که می‌توان با یک گره لبه‌ای واحد (Orange Pi) هم‌زمان وب‌سرویس امن، تشخیص رویداد بلاذرنگ و مانیتورینگ منابع را پیاده‌سازی کرد و با اتصال به یک بروکر مرکزی مطمئن، به پایداری عملیاتی، امنیت ارتباطی و انعطاف توسعه دست یافت. این معماری، علاوه بر پوشش تمام الزامات اصلی، با تحقق گزینه‌های امتیازی API، AJAX، Docker، RBAC و.. الگویی کاربردی و قابل تعمیم برای پروژه‌های صنعتی آموزشی IoT ارائه می‌کند.

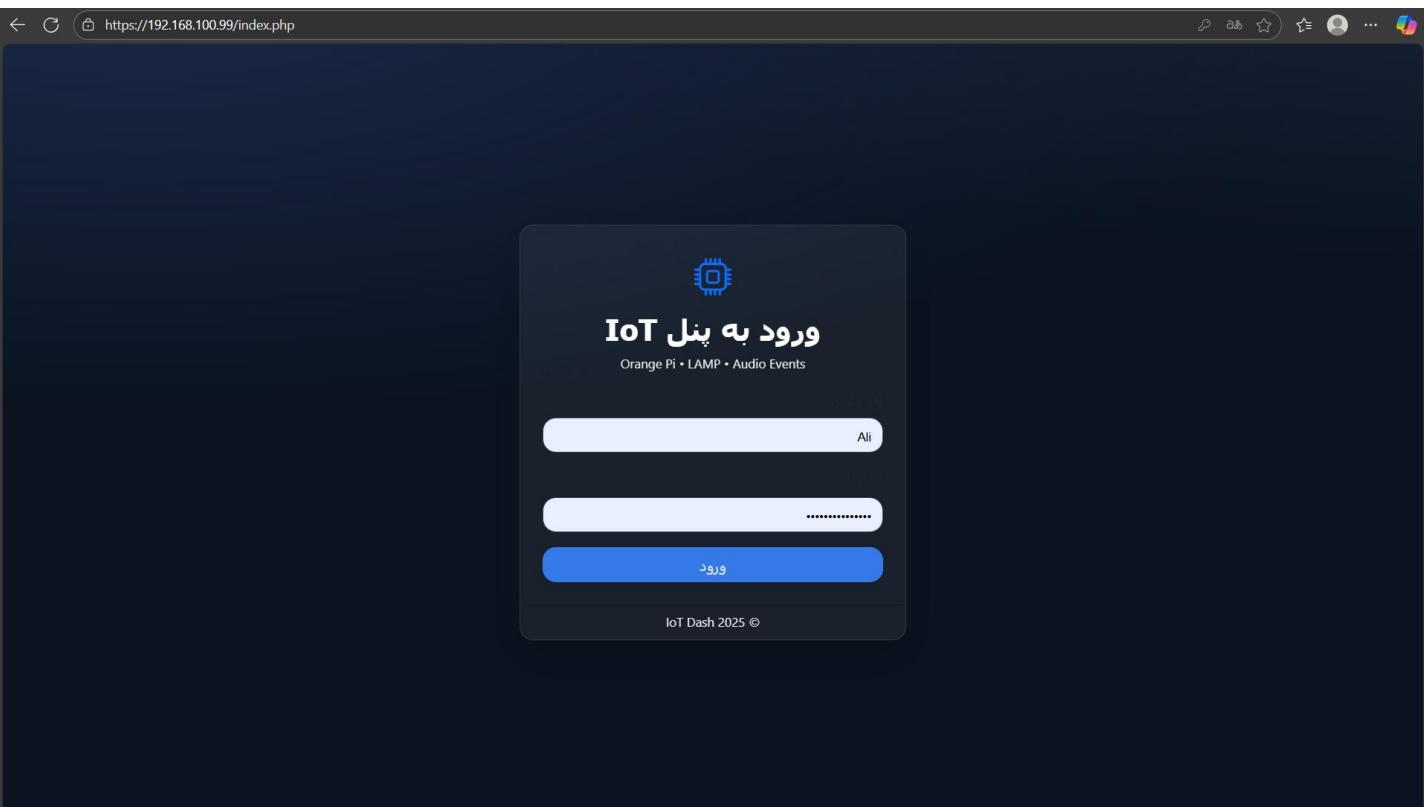
این پیاده‌سازی تمامی بندهای هسته‌ای طرح DET-AUDIO، DB-SEED، WEB-DASH، BASE-LAMP، BASE-OS، API-REST، MQTT-TLS، MQTT-PIPE، SVC-SYSTEMD، DB-ML-ADV، CNT-DOCKER، WEB-AJAX، INDEX را به طور کامل پوشش می‌دهد و فیچرهای امتیازی این ترتیب، پروژه نمونه‌ای عملی از طراحی معماری منطقی، امنیت مؤثر، پایداری عملیاتی و کیفیت ارائه می‌کند که برای کاربردهای صنعتی آموزشی IoT قابل تعمیم است.

اتصال میکروفون USB به Orange Pi



صفحه لاگین به سایت با نام کاربری و پسورد مخصوص اشخاص:

<https://192.168.100.99/iotdash/dashboard.php>

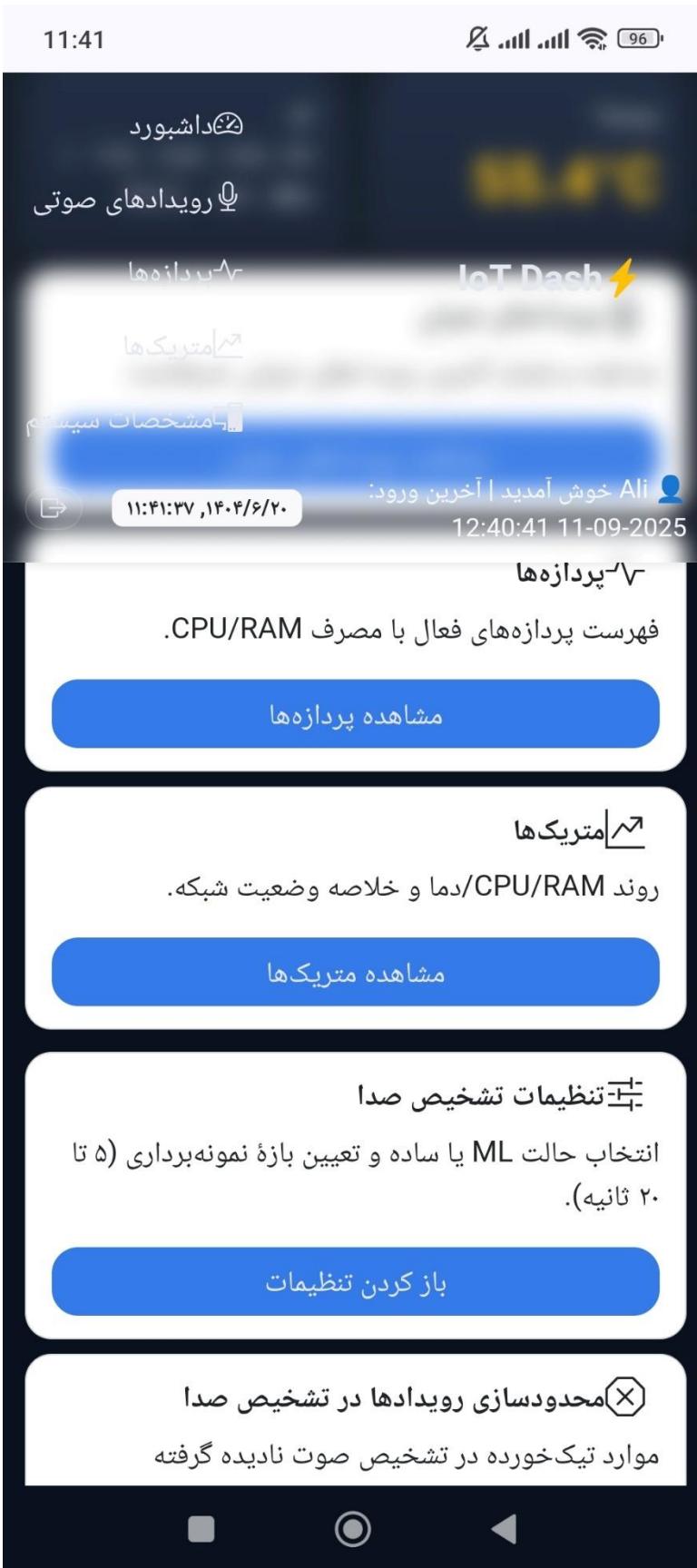


صفحه اولیه سایت :

The dashboard page displays various system metrics and control buttons. At the top, there's a header with the URL "192.168.100.99/dashboard.php", the date and time "12:03:42 11-09-2025", and a "Logout" button. The main area has four large status cards: IP (RSSI -63 dBm / 192.168.100.99), Temp (51.6°C), RAM (78.4%), and CPU (13.8%). Below these are six smaller cards: 1) "متربکها" (Network status) with a "مشاهده متربکها" button; 2) "پردازهها" (Processor status) with a "مشاهده پردازهها" button; 3) "رویدادهای صوتی" (Sound events) with a "مشاهده رویدادهای صوتی" button; 4) "تنظیمات تشخیص صدا" (Sound detection settings) with a "بار کردن تنظیمات" button; 5) "محدودسازی رویدادها در تشخیص صدا" (Restrict events in sound detection) with a "تنظیم محدودسازی" button; and 6) "نتایج حالت ML" (ML status results) with a "نمایش نتایج" button.

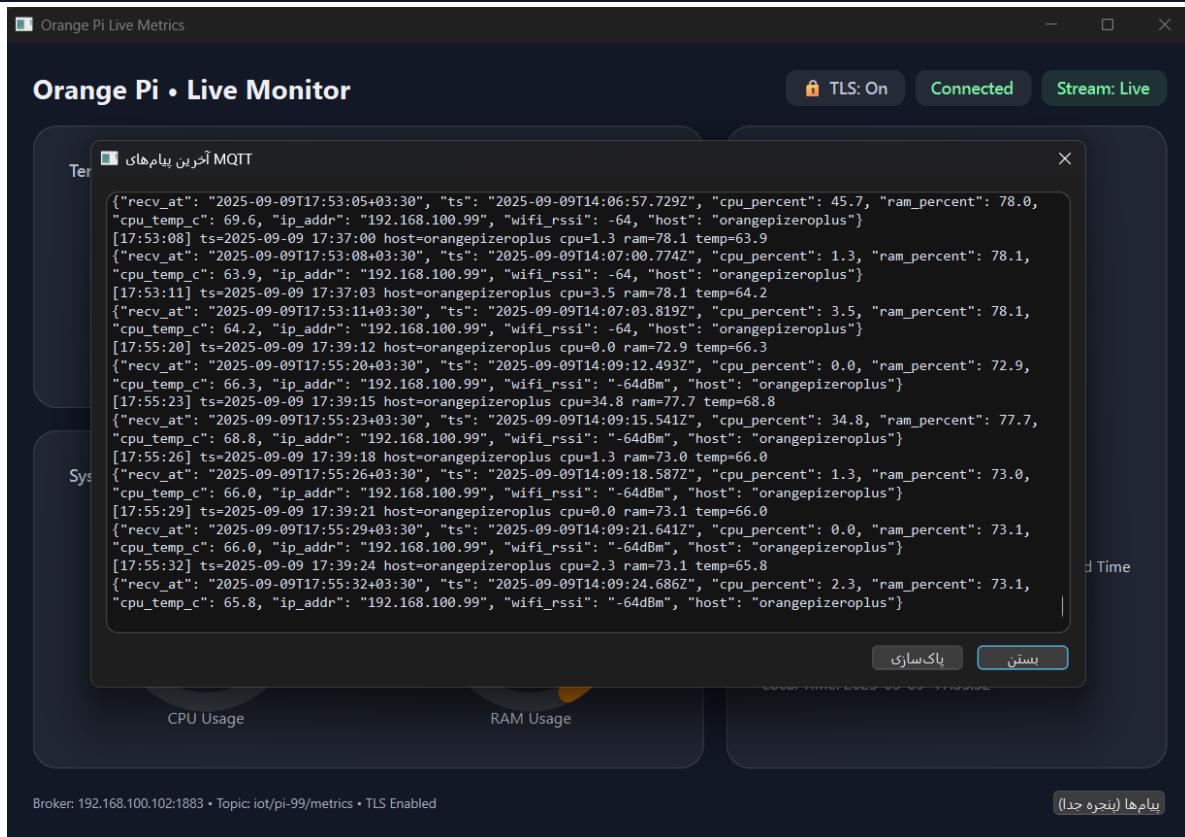
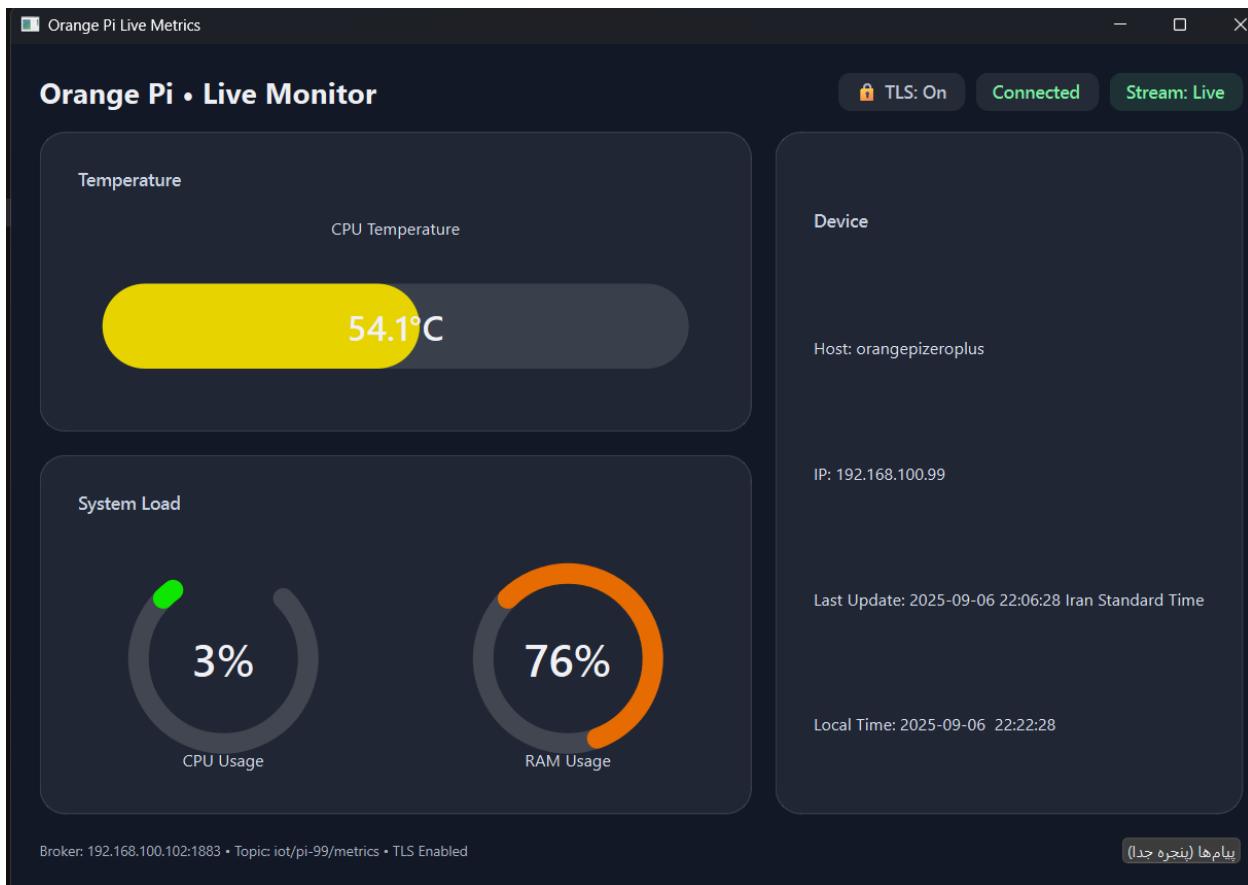
mobile readability:







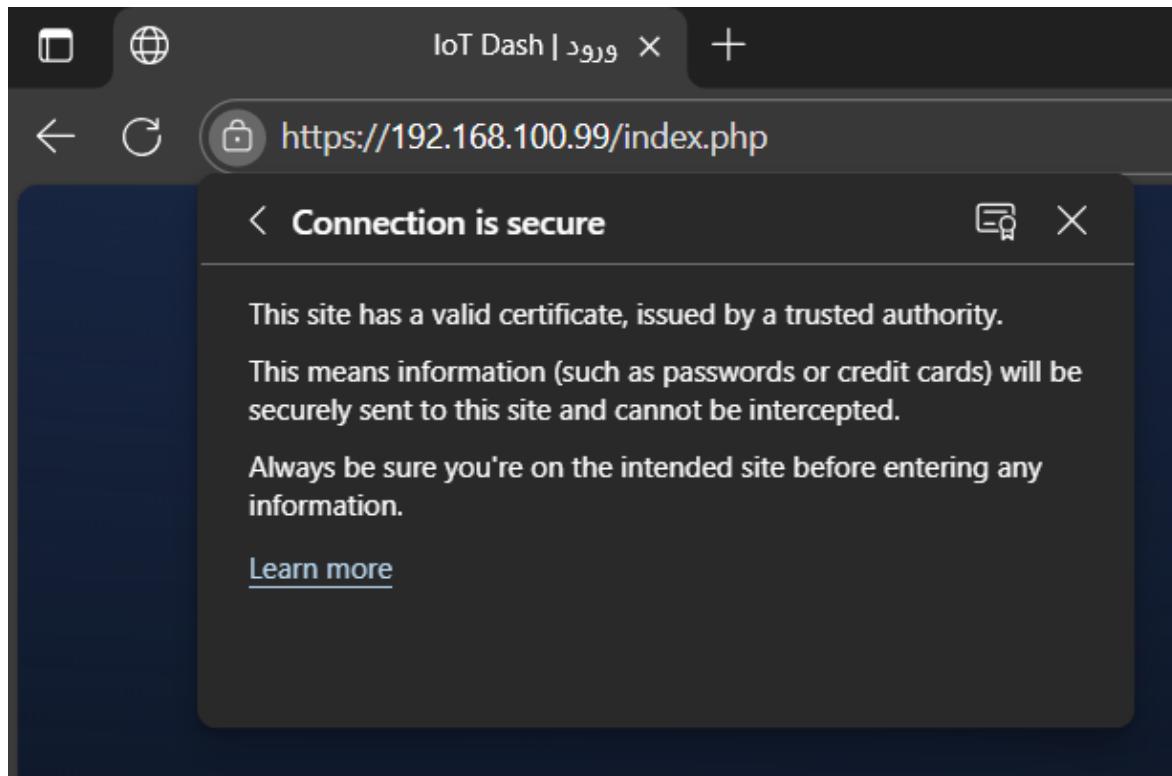
GUI طراحی شده برای ساپسکرایب تاپیک برای ACL و بروکر Mosquitto با TLS و MQTT :



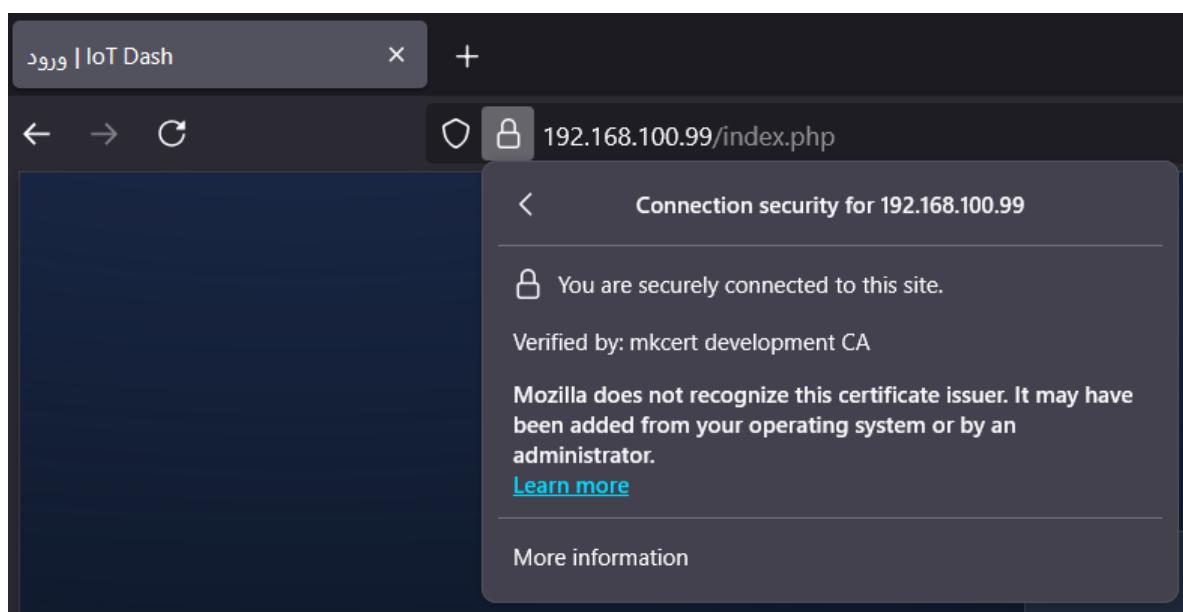
همانطور که مشخص است سایت دارای امنیت است وجود <https://> یعنی سرویس روی پورت 443 با SSL/TLS کار می‌کند.

و سایت دارای گواهی و تبادل کلید خصوصی و بین کامپیوتر و Orange Pi است که سایت را ایمن میکند

:Tایید امنیت سایت در مرورگر Edge



:Firefox



در مروگر کروم :

The screenshot shows a web browser window with the URL `192.168.100.99/dashboard.php`. A security overlay is displayed, stating "Connection is secure" and "Certificate is valid". Below this, a "Certificate Viewer" dialog is open, showing details for a certificate issued by "mkcert development CA" to "mkcert development certificate". The "General" tab is selected, displaying information such as the common name (CN) being "<Not Part Of Certificate>", the organization (O) being "mkcert development certificate", and the organizational unit (OU) being "LAPTOP-JR86J3E9\Ali@LAPTOP-JR86J3E9". The dialog also shows the validity period from September 6, 2025, to December 6, 2027, and SHA-256 fingerprints for both the certificate and public key.

Security
192.168.100.99

Connection is secure
Your information (for example, passwords or credit card numbers) is private when it is sent to this site. [Learn more](#)

Certificate is valid

Certificate Viewer: OU=LAPTOP-JR86J3E9\\Ali@LAPTOP-JR86J3E9,O=mkcert development certificate

General Details

Issued To

Common Name (CN)	<Not Part Of Certificate>
Organization (O)	mkcert development certificate
Organizational Unit (OU)	LAPTOP-JR86J3E9\Ali@LAPTOP-JR86J3E9

Issued By

Common Name (CN)	mkcert LAPTOP-JR86J3E9\Ali@LAPTOP-JR86J3E9
Organization (O)	mkcert development CA
Organizational Unit (OU)	LAPTOP-JR86J3E9\Ali@LAPTOP-JR86J3E9

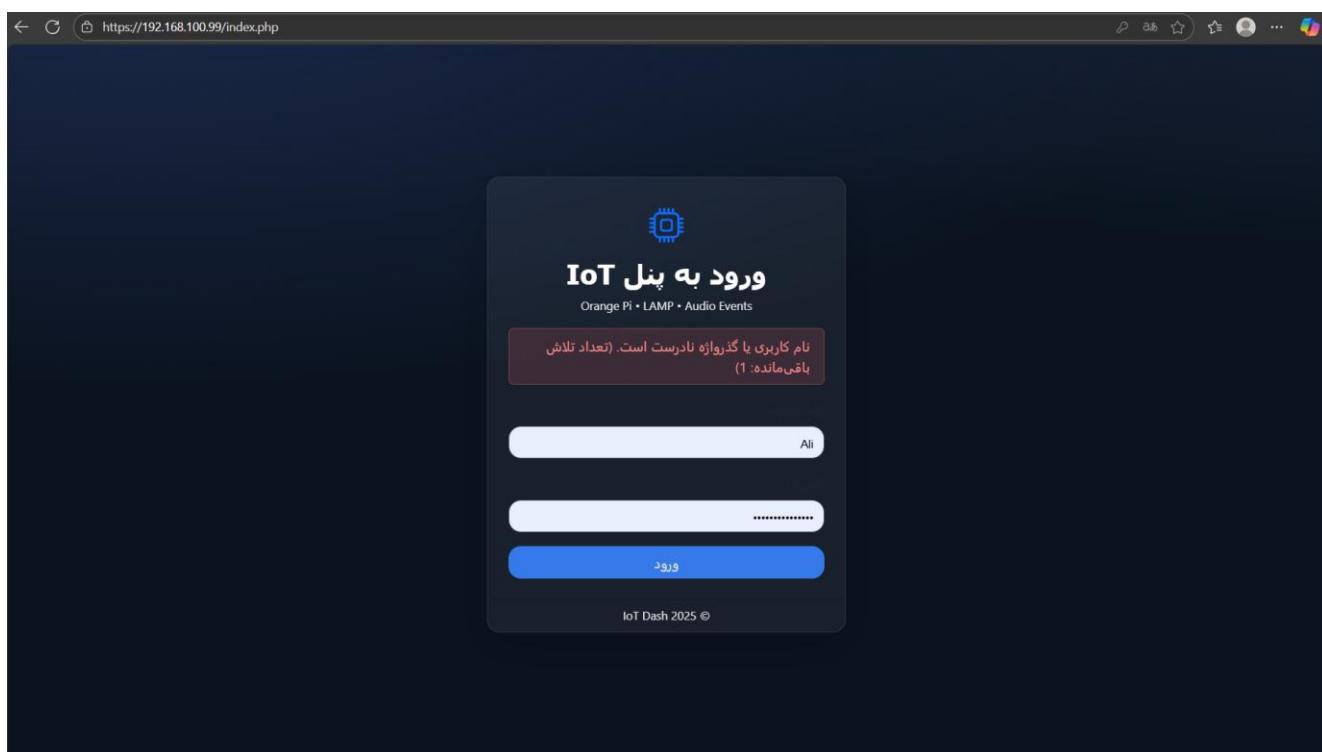
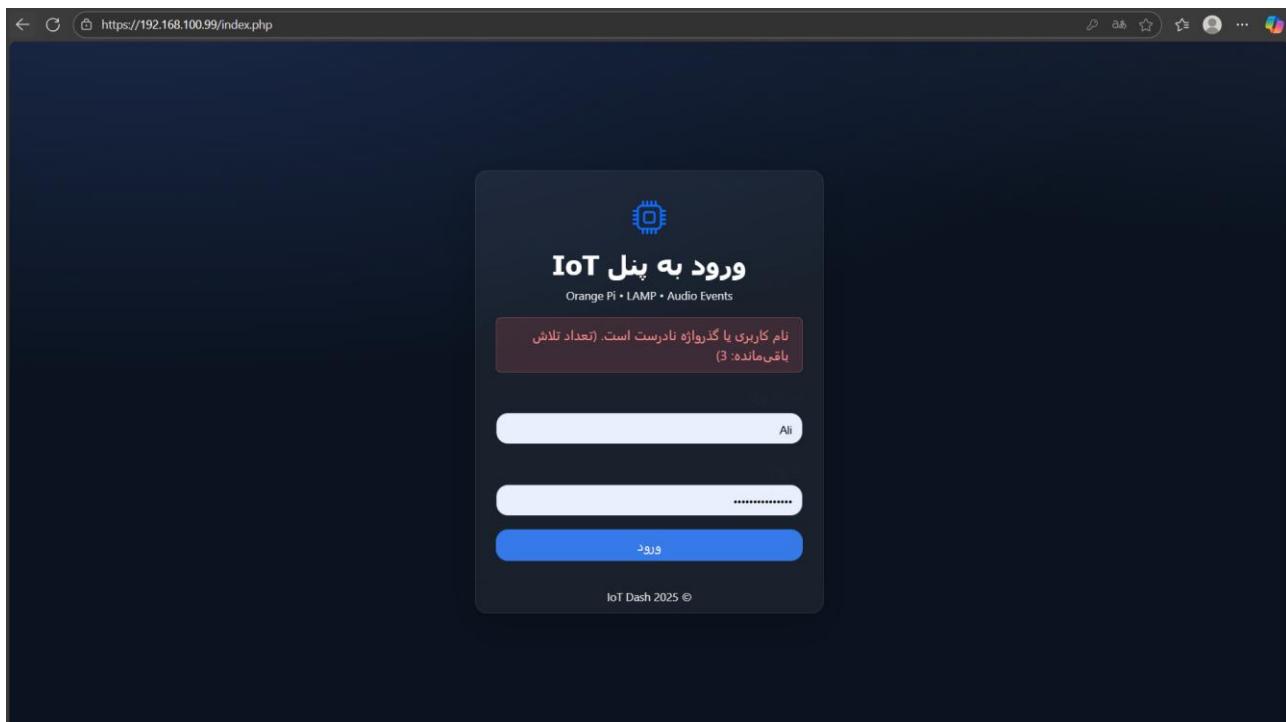
Validity Period

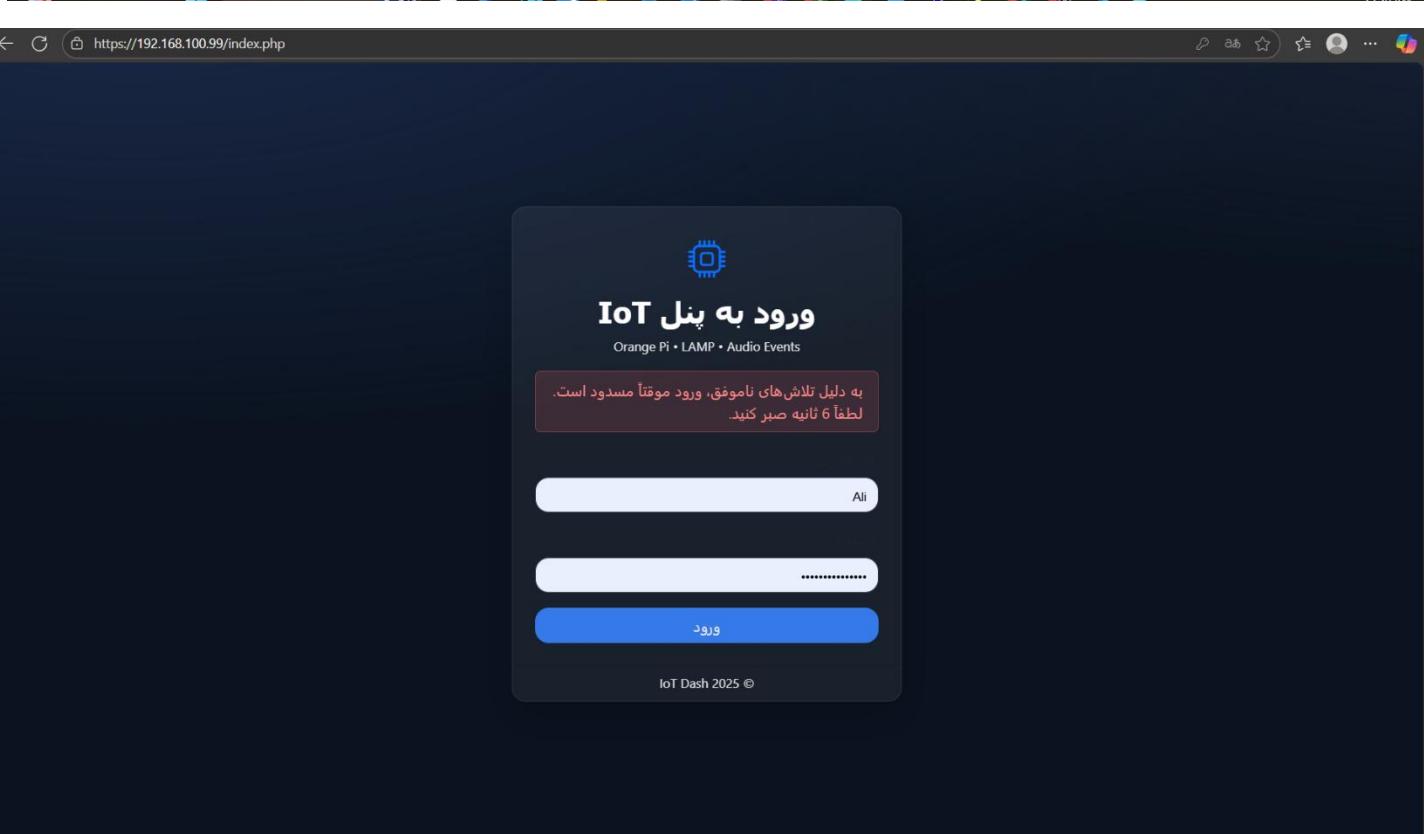
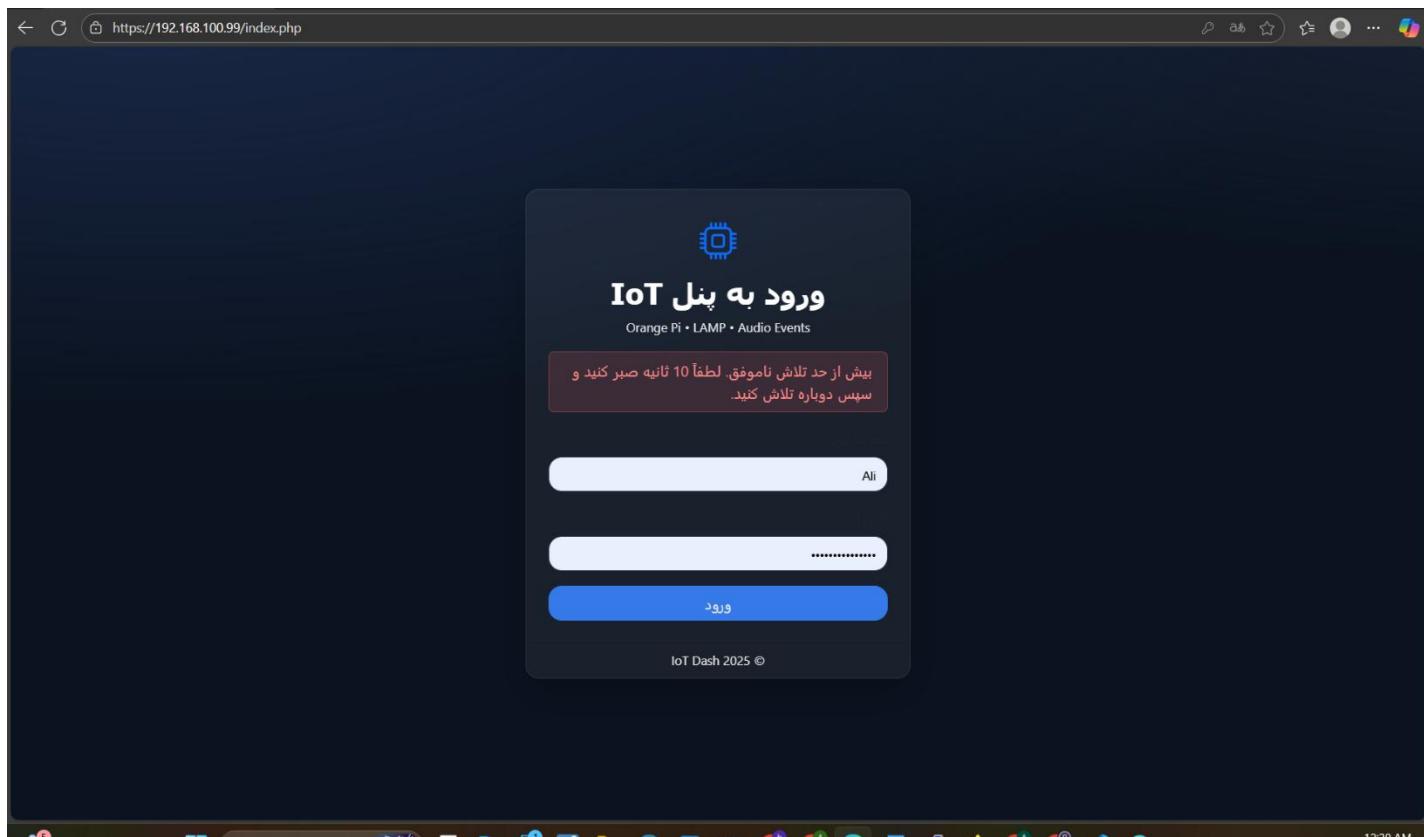
Issued On	Saturday, September 6, 2025 at 11:14:37 AM
Expires On	Monday, December 6, 2027 at 11:14:37 AM

SHA-256 Fingerprints

Certificate	b6925680491abff7b19b9641798504f0bda8fbf5806093888e19b69ef650271b
Public Key	2d601f8e331d4cc25510c9dd6f271a0921f11a3cba3ab830f17df8ac0d4c df99

همینطور با تنظیماتی که انجام دادیم سایت دارای استاتیک IP است یعنی فقط با یک IP مشخص 192.168.100.99 بالا می‌آید و دارای فایروال UFW است و کسی به سادگی نمیتواند به سایت نفوذ کندر صفحه ورود اگر رمز را اشتباه بزنیم نمیتوانیم ورود کنیم و سایت اخطار میدهد با هر بار اشتباه زدن رمز سایت میگوید فلان بار اشتباه رمز را وارد کردی تا وقتی که 4 بار پشت سر هم رمز را اشتباه بزنیم و سایت 10 ثانیه قفل کند به اصطلاح **Light rate-limiting** اتفاق میوقتد عکس :





در بخش رویداد های صوتی :

#	زمان	از	تا	برجسب / که	جستجو	داداگه ای صوتی	مشخصات سیستم	امنیتیک ها	رویداد های صوتی	ادمین	داداگه ای
1	۱۹:۰۱:۱۸ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
2	۱۹:۰۱:۱۹ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
3	۱۹:۰۱:۲۰ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
4	۱۹:۰۱:۲۱ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
5	۱۹:۰۱:۲۲ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
6	۱۹:۰۱:۲۳ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
7	۱۹:۰۱:۲۴ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
8	۱۹:۰۱:۲۵ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
9	۱۹:۰۱:۲۶ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
10	۱۹:۰۱:۲۷ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
11	۱۹:۰۱:۲۸ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
12	۱۹:۰۱:۲۹ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
13	۱۹:۰۱:۳۰ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
14	۱۹:۰۱:۳۱ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
15	۱۹:۰۱:۳۲ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
16	۱۹:۰۱:۳۳ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
17	۱۹:۰۱:۳۴ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
18	۱۹:۰۱:۳۵ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
19	۱۹:۰۱:۳۶ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
20	۱۹:۰۱:۳۷ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
21	۱۹:۰۱:۳۸ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
22	۱۹:۰۱:۳۹ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
23	۱۹:۰۱:۴۰ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										
24	۱۹:۰۱:۴۱ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵										

همینطور پایین سمت چپ سایت عملکرد و تایمینگ دیتابیس را بررسی میکند یعنی ایندکس گذاری روی جدول رویدادها برای بهبود کارایی کوئری ها.

Indexing and simple benchmark for query performance

عکس :

Ali-Node	86.0%	Silence	۱۹:۰۱:۰۰:۰۰ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	48
Ali-Node	76.0%	Silence	۱۹:۰۱:۰۰:۳۳ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	49
Ali-Node	96.0%	Silence	۱۹:۰۱:۰۰:۴۴ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	50

مجموع: 710 (سقف نمایش: 1000000) · عملکرد کوئری: ms 38 · کل = 5 ms

صفحه 1 از 15

تعداد در هر صفحه:

50

مجموع: 714 (سقف نمایش: 1000000) · عملکرد کوئری: ms 22 · DB=5 ms

صفحه 6 از 29

25

تعداد در هر صفحه:

25

تعداد رویداد در هر صفحه و عقب جلو کردن رویداد ها هم قابل تنظیم است:

Ali-Node	87.0%	Silence	۱۹:۰۱:۱۱:۱۱ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	148
Ali-Node	79.0%	Silence	۱۹:۰۱:۱۱:۲۵ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	149
Ali-Node	89.0%	Silence	۱۹:۰۱:۱۱:۵۰ ,۱۷:۵۷:۰۷/۰۹/۲۰۲۵	150

مجموع: 714 (سقف نمایش: 1000000) · عملکرد کوئری: ms 22 · DB=5 ms

صفحه 6 از 29

25

تعداد در هر صفحه:

25

مجموع: 714 (سقف نمایش: 1000000) · عملکرد کوئری: ms 22 = DB=5 ms · کل = 5 ms

در مورد عملکرد دیتابیس و کوئری :

DB=5 ms يعني خود کوئری دیتابیس SELECT روی جدول رویدادها فقط 5 میلی ثانیه طول کشیده.

کل 22 ms یعنی کل رفت و برگشت درخواست از لحظه‌ای که مرورگر fetch زده تا زمانی که جواب JSON امده و جدول رندر شده، 22 میلی ثانیه طول کشیده.

این شامل:

ارسال درخواست به سرور

اجرای PHP و کوئری DB

قابلیت سرج کردن رویداد بر اساس زمان و ساعت و نود ارسال کننده و اسم خود رویداد یا شدت اطمینانش رو داریم برای مثال :

سرج بر اساس اسم :

The screenshot shows a web browser displaying the IoT Dash dashboard at the URL <https://192.168.100.99/events.php>. The dashboard has a header with various system status indicators and navigation links. Below the header is a search bar with the query "music". The main content area is a table titled "جدائل اعتماد (%) حداکثر اعتماد (%)" (Dependence Schedules (%)) with columns: #, زمان (Time), برچسب (Label), اعتماد (Confidence), and گره (Node). The table lists 11 rows of data, all from the node "Ali-Node", with confidence values ranging from 71.0% to 97.0%. At the bottom of the table, it says "تعداد در هر صفحه: 25" (Number per page: 25) and "صفحه 1 از 1" (Page 1 of 1).

#	زمان	برچسب	اعتماد	گره
1	۱۸:۵۹:۳۰ ,۱۴۰۷/۷/۱۷	Music	83.0%	Ali-Node
2	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	96.0%	Ali-Node
3	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	76.0%	Ali-Node
4	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	90.0%	Ali-Node
5	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	97.0%	Ali-Node
6	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	79.0%	Ali-Node
7	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	89.0%	Ali-Node
8	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	71.0%	Ali-Node
9	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	87.0%	Ali-Node
10	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	86.0%	Ali-Node
11	۱۰:۵۹:۴۰ ,۱۴۰۷/۷/۱۷	Music	86.0%	Ali-Node

سرج بر اساس درصد اطمینان:

192.168.100.99/events.php 01:30:34 09-09-2025 Ali خوش آمدید | آخرین ورود: ۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸

داشبورد رویدادهای صوتی مشخصات سیستم امنیتیکها بردازهای امنیتیکها امنیتیکها داشبورد

IoT Dash

جدول اعتماد (%) حداکثر اعتماد (%)

اعتماد
کاربر
زمان

گردها	اعتماد	برچسب	زمان	#
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	1
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	2
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	3
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	4
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	5
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	6
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	7
Ali-Node	95.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	8
Ali-Node	96.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	9
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	10
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	11
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	12
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	13
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	14
Ali-Node	96.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	15
Ali-Node	96.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	16
Ali-Node	95.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	17
Ali-Node	96.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	18
Ali-Node	96.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	19
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	20
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	21

192.168.100.99/events.php 01:30:34 09-09-2025 Ali خوش آمدید | آخرین ورود: ۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸

داشبورد رویدادهای صوتی مشخصات سیستم امنیتیکها بردازهای امنیتیکها داشبورد

IoT Dash

جدول اعتماد (%) حداکثر اعتماد (%)

اعتماد
کاربر
زمان

گردها	اعتماد	برچسب	زمان	#
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	1
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	2
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	3
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	4
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	5
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	6
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	7
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	8
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	9
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	10
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	11
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	12
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	13
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	14
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	15
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	16
Ali-Node	94.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	17
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	18
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	19
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	20
Ali-Node	93.0%	Silence	۰۱:۵۶:۳۲,۱۹۴۷/۰۹/۰۸	21

IoT Dash

#	زمان	بررسی	اعتماد	گرده	حداقل اعتماد (%) / حداقل اعتماد (%)
1	۹:۱۴:۳۹, ۱۴۰۷/۵/۱۸	Silence	90.0%	Ali-Node	90.0%
2	۹:۱۴:۳۷, ۱۴۰۷/۵/۱۸	Silence	73.0%	Ali-Node	73.0%
3	۹:۱۴:۳۶, ۱۴۰۷/۵/۱۸	Silence	93.0%	Ali-Node	93.0%
4	۹:۱۴:۳۵, ۱۴۰۷/۵/۱۸	Silence	83.0%	Ali-Node	83.0%
5	۹:۱۴:۳۴, ۱۴۰۷/۵/۱۸	Silence	75.0%	Ali-Node	75.0%
6	۹:۱۴:۳۰, ۱۴۰۷/۵/۱۸	Silence	79.0%	Ali-Node	79.0%
7	۹:۱۴:۲۹, ۱۴۰۷/۵/۱۸	Silence	76.0%	Ali-Node	76.0%
8	۹:۱۴:۲۸, ۱۴۰۷/۵/۱۸	Silence	92.0%	Ali-Node	92.0%
9	۹:۱۴:۲۷, ۱۴۰۷/۵/۱۸	Silence	90.0%	Ali-Node	90.0%
10	۹:۱۴:۲۶, ۱۴۰۷/۵/۱۸	Silence	92.0%	Ali-Node	92.0%
11	۹:۱۴:۲۵, ۱۴۰۷/۵/۱۸	Silence	93.0%	Ali-Node	93.0%
12	۹:۱۴:۲۴, ۱۴۰۷/۵/۱۸	Silence	90.0%	Ali-Node	90.0%
13	۹:۱۴:۲۳, ۱۴۰۷/۵/۱۸	Silence	93.0%	Ali-Node	93.0%
14	۹:۱۴:۲۲, ۱۴۰۷/۵/۱۸	Silence	82.0%	Ali-Node	82.0%
15	۹:۱۴:۲۱, ۱۴۰۷/۵/۱۸	Silence	73.0%	Ali-Node	73.0%
16	۹:۱۴:۲۰, ۱۴۰۷/۵/۱۸	Silence	76.0%	Ali-Node	76.0%
17	۹:۱۴:۱۹, ۱۴۰۷/۵/۱۸	Silence	94.0%	Ali-Node	94.0%
18	۹:۱۴:۱۸, ۱۴۰۷/۵/۱۸	Silence	70.0%	Ali-Node	70.0%
19	۹:۱۴:۱۷, ۱۴۰۷/۵/۱۸	Silence	94.0%	Ali-Node	94.0%
20	۹:۱۴:۱۶, ۱۴۰۷/۵/۱۸	Silence	76.0%	Ali-Node	76.0%

سرچ بر اساس تاریخ رویداد: (قابل تنظیم است که میلادی سرج کند یا شمسی بر اساس تاریخ سیستم پیشنهاد میدهد)

IoT Dash

#	زمان	بررسی	اعتماد	گرده	حداقل اعتماد (%) / حداقل اعتماد (%)
1	۹:۱۴:۳۹, ۱۴۰۷/۵/۱۸	Silence	75.0%	Ali-Node	95.0%
2	۹:۱۴:۳۷, ۱۴۰۷/۵/۱۸	Silence	87.0%	Ali-Node	95.0%
3	۹:۱۴:۳۶, ۱۴۰۷/۵/۱۸	Silence	93.0%	Ali-Node	95.0%
4	۹:۱۴:۳۵, ۱۴۰۷/۵/۱۸	Silence	87.0%	Ali-Node	95.0%
5	۹:۱۴:۳۴, ۱۴۰۷/۵/۱۸	Silence	91.0%	Ali-Node	95.0%
6	۹:۱۴:۳۳, ۱۴۰۷/۵/۱۸	Silence	90.0%	Ali-Node	95.0%
7	۹:۱۴:۳۲, ۱۴۰۷/۵/۱۸	Silence	73.0%	Ali-Node	95.0%

September 2025	↑	↓				
Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11
Clear			Today			

داشبورد رویدادهای صوتی پردازهای متریک‌ها مشخصات سیستم Ali جوش آمدید | آخرین ورود 01:30:34 09-09-2025

حداقل اعتماد (%) حداکثر اعتماد (%)

برچسب جستجو تاریخ از

#	زمان	برچسب	جستجو	تاریخ	از
1	۹:۱۳:۴۷, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۴۷, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
2	۹:۱۳:۴۹, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۴۹, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
3	۹:۱۳:۵۰, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۰, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
4	۹:۱۳:۵۱, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۱, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
5	۹:۱۳:۵۳, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۳, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
6	۹:۱۳:۵۴, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۴, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
7	۹:۱۳:۵۵, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۵, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
8	۹:۱۳:۵۶, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۶, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
9	۹:۱۳:۵۷, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۷, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
10	۹:۱۳:۵۸, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۸, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
11	۹:۱۳:۵۹, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۵۹, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
12	۹:۱۳:۶۰, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۰, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
13	۹:۱۳:۶۱, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۱, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
14	۹:۱۳:۶۲, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۲, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
15	۹:۱۳:۶۳, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۳, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
16	۹:۱۳:۶۴, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۴, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
17	۹:۱۳:۶۵, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۵, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
18	۹:۱۳:۶۶, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۶, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
19	۹:۱۳:۶۷, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۷, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM
20	۹:۱۳:۶۸, ۰۹-۰۹-۲۰۲۵	Silence	۹:۱۳:۶۸, ۰۹-۰۹-۲۰۲۵	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ PM

اعمال همه‌ی گذینه‌های سرج با هم :

داشبورد رویدادهای صوتی پردازهای متریک‌ها مشخصات سیستم Ali جوش آمدید | آخرین ورود 01:30:34 09-09-2025

حداقل اعتماد (%) حداکثر اعتماد (%)

برچسب جستجو تاریخ از

#	زمان	برچسب	جستجو	تاریخ	از
1	۱۰:۴۰:۰۸, ۰۹-۰۹-۲۰۲۵	Police car (siren)	car	۰۹/۰۹/۲۰۲۵ ۱۲:۵۹ PM	۰۹/۰۸/۲۰۲۵ ۱۲:۵۹ AM

ms 36 ms - کل = 1000000 (سقف نمایش: 1) · عملکرد کوئری: DB=12 ms

تعداد در هر صفحه: 25

صفحه‌ی پردازه‌ها :

The screenshot shows a web-based process monitoring interface titled "IoT Dash". The main content is a table listing system processes. The columns are: #, PID, کاربر (User), نام (Name), %CPU, %RAM, زمان (Time), and عملیات (Operations). The table contains 20 rows of data.

#	PID	کاربر	نام	%CPU	%RAM	زمان	عملیات
1	1319	root	python3	48.8	30.8	۹:۰۷:۳۱ ,۱۴۰۴/۶/۱۸	(X)
2	1	root	systemd	1	1.7	۹:۰۷:۰۱ ,۱۴۰۴/۶/۱۸	(X)
3	1122	root	python3	2.7	0.9	۹:۰۷:۱۸ ,۱۴۰۴/۶/۱۸	(X)
4	570	root	systemd-journal	0.9	0.7	۹:۰۷:۱۹ ,۱۴۰۴/۶/۱۸	(X)
5	156	root	kswapd0	0	0.6	۹:۰۷:۰۲ ,۱۴۰۴/۶/۱۸	(X)
6	614	+message	dbus-daemon	0.4	0.5	۹:۰۷:۱۰ ,۱۴۰۴/۶/۱۸	(X)
7	1245	mysql	mysqld	3.5	0.5	۹:۰۷:۱۹ ,۱۴۰۴/۶/۱۸	(X)
8	230	root	kworker/3:1H-mm	0	0.2	۹:۰۷:۰۰ ,۱۴۰۴/۶/۱۸	(X)
9	324	root	systemd-udevd	0.6	0.2	۹:۰۷:۰۹ ,۱۴۰۴/۶/۱۸	(X)
10	574	root	haveged	0.5	0.2	۹:۰۷:۱۹ ,۱۴۰۴/۶/۱۸	(X)
11	619	root	NetworkManager	1.5	0.2	۹:۰۷:۱۰ ,۱۴۰۴/۶/۱۸	(X)
12	654	syslog	rsyslogd	0.3	0.2	۹:۰۷:۱۰ ,۱۴۰۴/۶/۱۸	(X)
13	658	root	systemd-logind	0.7	0.2	۹:۰۷:۱۰ ,۱۴۰۴/۶/۱۸	(X)
14	530	root	ksdioirqd/mmc1	0	0.1	۹:۰۷:۱۳ ,۱۴۰۴/۶/۱۸	(X)
15	650	root	networkd-dispat	1	0.1	۹:۰۷:۱۰ ,۱۴۰۴/۶/۱۸	(X)
16	1159	root	unattended-upgr	2.3	0.1	۹:۰۷:۱۸ ,۱۴۰۴/۶/۱۸	(X)
17	2	root	kthreadd	0	0	۹:۰۷:۰۳ ,۱۴۰۴/۶/۱۸	(X)
18	3	root	rcu_gp	0	0	۹:۰۷:۰۰ ,۱۴۰۴/۶/۱۸	(X)
19	4	root	rcu_par_gp	0	0	۹:۰۷:۰۰ ,۱۴۰۴/۶/۱۸	(X)
20	7	root	kworker/u8:0-ev	0	0	۹:۰۷:۰۱ ,۱۴۰۴/۶/۱۸	(X)

همانطور که مشخص است این صفحه مانند Task Manager کامپیوتر برنامه‌های در حال اجرا را نمایش میدهد

دو یوزر تعریف کردیم (وب سایت قابلیت تعریف یوزر های مختلف با سطخ دسترسی های مختلفی را دارد)

یوزر اول

User : Ali

Pass : EmbeddedProject

که Ali همه‌ی سطح‌های دسترسی را دارا است

یوزر دوم

User : Hassan

Pass : 1234

ولی Hassan سطح دسترسی محدود تری دارد در صفحه‌ی پردازه‌ها ما قابلیت از بستن برنامه‌ها را داریم

تمامی قابلیت‌های وب سایت و تشخیص رویداد و درج در دیتا بیس و بروکر MQTT و Mosquitto همراه با TLS و WLT به صورت خودکار در root اجرا میگردد و با هربار ریست شدن سیستم یا بالا آمدن سیستم به صورت خودکار اجرا میشوند برای اجرای برنامه‌های دیگر یک کاربر دیگر به نام ali در Orange Pi تعریف کردیم یعنی root فقط برای سیستم اصلی ما که تشخیص رویداد است کار میکند و کسی اجازه ورود به root را ندارد و ورود به root غیرفعال است

حال اگر کاربر ali در Orange Pi برنامه‌ای اجرا کند

کاربر Ali در صفحه پردازه‌ها قابلیت Kill کردن یا بستن آن‌ها را دارد ولی Hassan نمیتواند توجه شوند که ali در Orange Pi و در وب سایت Ali کاربر‌های متفاوتی هستند و ربطی به هم ندارند

برای مثال :

کاربر ali در Orange Pi برنامه را اجرا میکند :

```
COM3 - PuTTY

Login incorrect
orangepi@orangepiplus: login: ali
Password:
[REDACTED]
Welcome to Orange Pi Focal with Linux 5.4.65-sunxi64

System load: 2.42 0.82 0.29 Up time: 1 min
Memory usage: 65 % of 477MB IP: 192.168.100.99
CPU temp: 35°C
Usage of /: 30% of 29G

[ General system configuration (beta): orangepi-config ]

Last login: Sat Sep 6 18:34:39 UTC 2025 from 192.168.100.102 on pts/0
ali@orangepiplus:~$ cd /opt/iot/python
ali@orangepiplus:/opt/iot/python$ python3 run.py
```

کاربر Ali در سایت میتواند برنامه را ببند :

192.168.100.99/processes.php

https://192.168.100.99/processes.php

13:29:11 09-09-2025 | آریان ورد | Ali جوش آمدید | آخرین ورود: 13:29:11 09-09-2025

داشبورد رویدادهای صوتی پردازهای صوتی مشخصات سیستم امتریک ها بروزرسانی آخرین ورود: Ali

IoT Dash

عملیات	زمان	%RAM	%CPU	نام	کاربر	PID	#
(X)	۹:۳۶:۵۱, ۱۴۰۴/۶/۱۸	38.8	36.5	python3	root	1322	1
(X)	۹:۳۶:۵۲, ۱۴۰۴/۶/۱۸	1.4	6	python3	ali	1840	2
(X)	۹:۳۶:۵۳, ۱۴۰۴/۶/۱۸	0.8	4.1	systemd	root	1	3
(X)	۹:۳۶:۵۴, ۱۴۰۴/۶/۱۸	0	3.4	kswapd0	root	156	4
(X)	۹:۳۶:۵۵, ۱۴۰۴/۶/۱۸	2	1.4	python3	root	1124	5
(X)	۹:۳۶:۵۶, ۱۴۰۴/۶/۱۸	0.3	1.1	systemd-udevd	root	326	6
(X)	۹:۳۶:۵۷, ۱۴۰۴/۶/۱۸	0.2	1.1	haveged	root	575	7
(X)	۹:۳۶:۵۸, ۱۴۰۴/۶/۱۸	0.6	1	systemd-journal	root	572	8
(X)	۹:۳۶:۵۹, ۱۴۰۴/۶/۱۸	0.9	1	NetworkManager	root	611	9
(X)	۹:۳۶:۶۰, ۱۴۰۴/۶/۱۸	0.3	0.8	dbus-daemon	+message	606	10
(X)	۹:۳۶:۶۱, ۱۴۰۴/۶/۱۸	9.9	0.7	mysqld	mysql	1247	11
(X)	۹:۳۶:۶۲, ۱۴۰۴/۶/۱۸	0.7	0.6	networkd-dispat	root	631	12
(X)	۹:۳۶:۶۳, ۱۴۰۴/۶/۱۸	1.3	0.5	unattended-upgr	root	1163	13
(X)	۹:۳۶:۶۴, ۱۴۰۴/۶/۱۸	0.8	0.5	systemd	ali	1731	14
(X)	۹:۳۶:۶۵, ۱۴۰۴/۶/۱۸	0.7	0.4	systemd-resolve	+systemd	576	15
(X)	۹:۳۶:۶۶, ۱۴۰۴/۶/۱۸	0.6	0.4	systemd-logind	root	654	16
(X)	۹:۳۶:۶۷, ۱۴۰۴/۶/۱۸	0.5	0.4	bash	ali	1738	17
(X)	۹:۳۶:۶۸, ۱۴۰۴/۶/۱۸	0	0.3	kworker/3:1H-mm	root	228	18
(X)	۹:۳۶:۶۹, ۱۴۰۴/۶/۱۸	0.4	0.3	wpa_supplicant	root	656	19
(X)	۹:۳۶:۷۰, ۱۴۰۴/۶/۱۸	0	0.2	rcu_sched	root	10	20

192.168.100.99/processes.php

https://192.168.100.99/processes.php

13:29:11 09-09-2025 | آریان ورد | Ali جوش آمدید | آخرین ورود: 13:29:11 09-09-2025

داشبورد آغازینه بردازه 1840 (ali) مطمئن هستند

192.168.100.99 says

OK Cancel

عملیات	زمان	%RAM	%CPU	نام	کاربر	PID	#
(X)	۹:۳۶:۵۱, ۱۴۰۴/۶/۱۸	38.8	34.9	python3	root	1322	1
(X)	۹:۳۶:۵۲, ۱۴۰۴/۶/۱۸	0.8	4	systemd	root	1	2
(X)	۹:۳۶:۵۳, ۱۴۰۴/۶/۱۸	0	3.2	kswapd0	root	156	3
(X)	۹:۳۶:۵۴, ۱۴۰۴/۶/۱۸	1.4	2.2	python3	ali	1840	4
(X)	۹:۳۶:۵۵, ۱۴۰۴/۶/۱۸	2	1.4	python3	root	1124	5
(X)	۹:۳۶:۵۶, ۱۴۰۴/۶/۱۸	0.3	1	systemd-udevd	root	326	6
(X)	۹:۳۶:۵۷, ۱۴۰۴/۶/۱۸	0.6	1	systemd-journal	root	572	7
(X)	۹:۳۶:۵۸, ۱۴۰۴/۶/۱۸	0.2	1	haveged	root	575	8
(X)	۹:۳۶:۵۹, ۱۴۰۴/۶/۱۸	0.9	0.9	NetworkManager	root	611	9
(X)	۹:۳۶:۶۰, ۱۴۰۴/۶/۱۸	0.3	0.8	dbus-daemon	+message	606	10
(X)	۹:۳۶:۶۱, ۱۴۰۴/۶/۱۸	9.9	0.7	mysqld	mysql	1247	11
(X)	۹:۳۶:۶۲, ۱۴۰۴/۶/۱۸	0.7	0.6	networkd-dispat	root	631	12
(X)	۹:۳۶:۶۳, ۱۴۰۴/۶/۱۸	1.3	0.5	unattended-upgr	root	1163	13
(X)	۹:۳۶:۶۴, ۱۴۰۴/۶/۱۸	0.7	0.4	systemd-resolve	+systemd	576	14
(X)	۹:۳۶:۶۵, ۱۴۰۴/۶/۱۸	0.6	0.4	systemd-logind	root	654	15
(X)	۹:۳۶:۶۶, ۱۴۰۴/۶/۱۸	0.8	0.4	systemd	ali	1731	16
(X)	۹:۳۶:۶۷, ۱۴۰۴/۶/۱۸	0.5	0.4	bash	ali	1738	17
(X)	۹:۳۶:۶۸, ۱۴۰۴/۶/۱۸	0	0.3	kworker/3:1H-mm	root	228	18
(X)	۹:۳۶:۶۹, ۱۴۰۴/۶/۱۸	0.4	0.3	wpa_supplicant	root	656	19
(X)	۹:۳۶:۷۰, ۱۴۰۴/۶/۱۸	0	0.2	rcu_sched	root	10	20

عملیات	زمان	نام	کاربر	PID	#
(X)	۹:۳۶:۵۰ ,۱۹۰۷۰۸/۱۸	38.7	35.4	python3	root 1
(X)	۹:۳۶:۵۸ ,۱۹۰۷۰۸/۱۸	0.8	3.8	systemd	root 2
(X)	۹:۳۶:۳۰ ,۱۹۰۷۰۸/۱۸	0	3	kswapd0	root 3
(X)	۹:۳۶:۵۰ ,۱۹۰۷۰۸/۱۸	2	1.3	python3	root 4
(X)	۹:۳۸:۴۲ ,۱۹۰۷۰۸/۱۸	1.4	1.1	python3	ali 1840
(X)	۹:۳۶:۳۴ ,۱۹۰۷۰۸/۱۸	0.3	1	systemd-udevd	root 326
(X)	۹:۳۶:۴۱ ,۱۹۰۷۰۸/۱۸	0.6	1	systemd-journal	root 572
(X)	۹:۳۶:۴۱ ,۱۹۰۷۰۸/۱۸	0.2	0.9	havaged	root 575
(X)	۹:۳۶:۴۷ ,۱۹۰۷۰۸/۱۸	0.9	0.9	NetworkManager	root 611
(X)	۹:۳۶:۴۷ ,۱۹۰۷۰۸/۱۸	0.3	0.8	dbus-daemon	+message 606
(X)	۹:۳۶:۴۶ ,۱۹۰۷۰۸/۱۸	9.9	0.6	mysqld	mysql 1247
(X)	۹:۳۶:۴۷ ,۱۹۰۷۰۸/۱۸	0.7	0.5	networkd-dispat	root 631
(X)	۹:۳۶:۴۰ ,۱۹۰۷۰۸/۱۸	1.3	0.5	unattended-upgr	root 1163
(X)	۹:۳۶:۴۰ ,۱۹۰۷۰۸/۱۸	0.7	0.4	systemd-resolve	+systemd 576
(X)	۹:۳۶:۴۰ ,۱۹۰۷۰۸/۱۸	0.6	0.4	systemd-logind	root 654
(X)	۹:۳۶:۳۰ ,۱۹۰۷۰۸/۱۸	0.8	0.4	systemd	ali 1731
(X)	۹:۳۶:۳۰ ,۱۹۰۷۰۸/۱۸	0	0.3	kworker/3:1H-mm	root 228
(X)	۹:۳۶:۴۷ ,۱۹۰۷۰۸/۱۸	0.4	0.3	wpa_supplicant	root 656
(X)	۹:۳۶:۳۰ ,۱۹۰۷۰۸/۱۸	0.5	0.3	bash	ali 1738
(X)	۹:۳۶:۴۷ ,۱۹۰۷۰۸/۱۸	0	0.2	rcu_sched	root 10

```
COM3 - PuTTY

Welcome to Orange Pi Focal with Linux 5.4.65-sunxi64

System load:  2.42 0.82 0.29   Up time:      1 min
Memory usage: 65 % of 477MB     IP:          192.168.100.99
CPU temp:    35°C
Usage of /:   30% of 29G

[ General system configuration (beta): orangepi-config ]

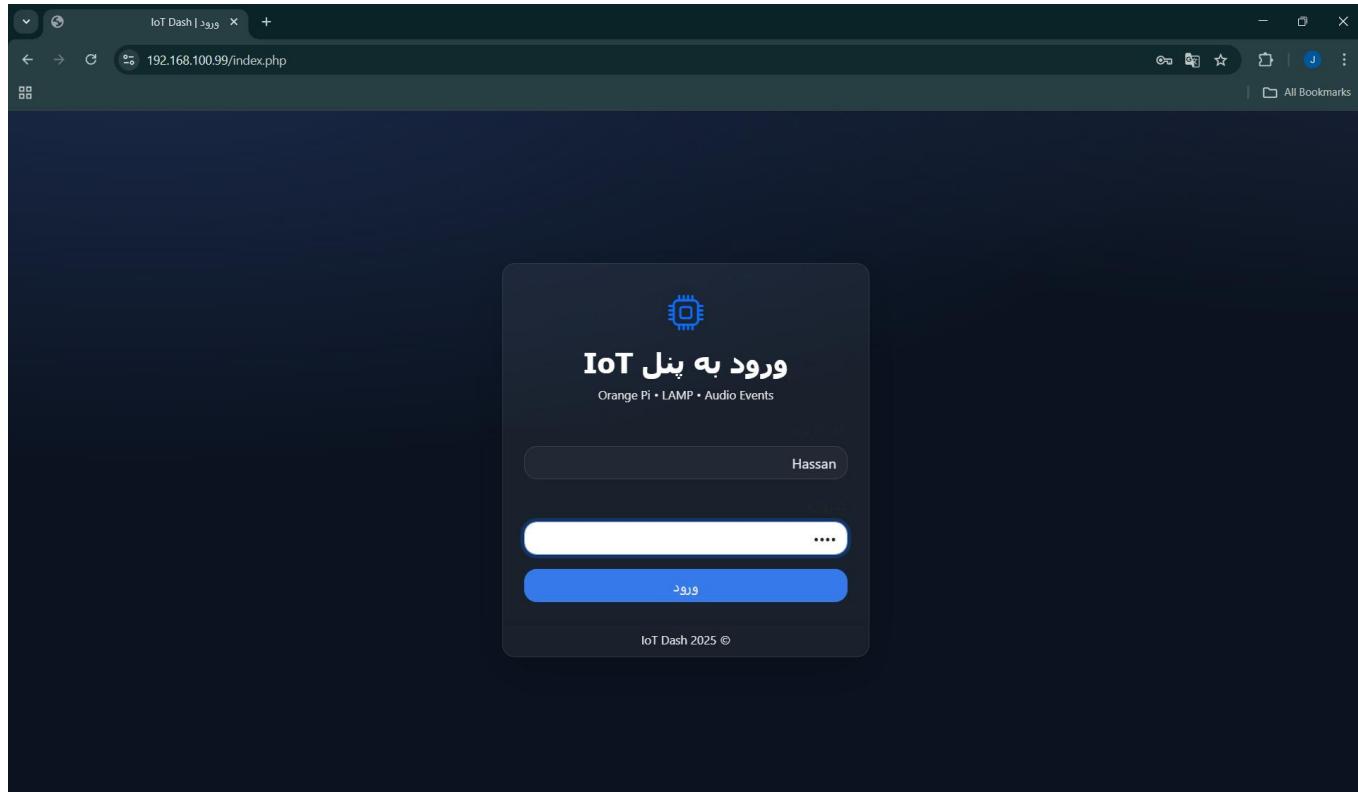
Last login: Sat Sep  6 18:34:39 UTC 2025 from 192.168.100.102 on pts/0

ali@orangepirzeroplus:~$ cd /opt/iot/python
ali@orangepirzeroplus:/opt/iot/python$ python3 run.py
^CTraceback (most recent call last):
  File "run.py", line 8, in <module>
    time.sleep(1)
KeyboardInterrupt

ali@orangepirzeroplus:/opt/iot/python$ python3 run.py

Terminated
```

ولی کاربر Hassan در وب سایت نمیتواند این کار را انجام دهد و به خطاب میخورد:



192.168.100.99/processes.php

192.168.100.99/processes.php

192.168.100.99 says

آبا از حاتمه بردازه 2483 (ali) مطمئن هستید؟

OK Cancel

#	PID	کاربر	نام
1	1322	root	python3
2	1	root	systemd
3	156	root	kswapd0
4	1124	root	python3
5	572	root	systemd-journal
6	606	+message	dbus-daemon
7	2483	ali	python3
8	1247	mysql	mysqld
9	326	root	systemd-udevd
10	575	root	haveged
11	611	root	NetworkManager
12	644	syslog	rsyslogd
13	654	root	systemd-logind
14	10	root	rcu_sched
15	228	root	kworker/3:1H-mm
16	508	root	ksdiorqd/mmc1
17	576	+systemd	systemd-resolve
18	631	root	networkd-dispat
19	1163	root	unattended-upgr

192.168.100.99/processes.php

19.168.100.99 says

denied_critical

خطا در IoT Dash

All Bookmarks

عملیات	زمان					
(X)	۹:۳۶:۵۱ ,۱۴*۵/۶/۲۰۲۵	۵۵.۰	۵۱.۵	python3	root	1322 1
(X)	۹:۳۶:۵۸ ,۱۴*۵/۶/۲۰۲۵	0.8	1.8	systemd	root	1 2
(X)	۹:۳۶:۵۰ ,۱۴*۵/۶/۲۰۲۵	0	0.9	kswapd0	root	156 3
(X)	۹:۳۶:۴۵ ,۱۴*۵/۶/۲۰۲۵	1.9	0.8	python3	root	1124 4
(X)	۹:۳۶:۴۱ ,۱۴*۵/۶/۲۰۲۵	0.7	0.7	systemd-journal	root	572 5
(X)	۹:۳۶:۴۳ ,۱۴*۵/۶/۲۰۲۵	0.3	0.5	dbus-daemon	+message	606 6
(X)	۹:۳۶:۴۶ ,۱۴*۵/۶/۲۰۲۵	9.1	0.3	mysqld	mysql	1247 7
(X)	۹:۳۶:۴۳ ,۱۴*۵/۶/۲۰۲۵	0.4	0.2	systemd-udevd	root	326 8
(X)	۹:۳۶:۴۴ ,۱۴*۵/۶/۲۰۲۵	0.2	0.2	haveged	root	575 9
(X)	۹:۳۶:۴۷ ,۱۴*۵/۶/۲۰۲۵	1.1	0.2	NetworkManager	root	611 10
(X)	۹:۳۶:۴۵ ,۱۴*۵/۶/۲۰۲۵	0.1	0.2	rsyslogd	syslog	644 11
(X)	۹:۳۶:۴۶ ,۱۴*۵/۶/۲۰۲۵	0.5	0.2	systemd-logind	root	654 12
(X)	۹:۳۶:۴۱ ,۱۴*۵/۶/۲۰۲۵	1.5	0.2	python3	ali	2483 13
(X)	۹:۳۶:۴۸ ,۱۴*۵/۶/۲۰۲۵	0	0.1	rcu_sched	root	10 14
(X)	۹:۳۶:۴۳ ,۱۴*۵/۶/۲۰۲۵	0	0.1	kworker/3:1H-mm	root	228 15
(X)	۹:۳۶:۴۸ ,۱۴*۵/۶/۲۰۲۵	0	0.1	ksdioirqq/mmc1	root	508 16
(X)	۹:۳۶:۴۵ ,۱۴*۵/۶/۲۰۲۵	0.7	0.1	systemd-resolve	+systemd	576 17
(X)	۹:۳۶:۴۵ ,۱۴*۵/۶/۲۰۲۵	0.7	0.1	networkd-dispat	root	631 18
(X)	۹:۳۶:۴۰ ,۱۴*۵/۶/۲۰۲۵	1.2	0.1	unattended-upgr	root	1163 19

صفحه‌ی متریک‌ها:

192.168.100.99/metrics.php

https://192.168.100.99/metrics.php

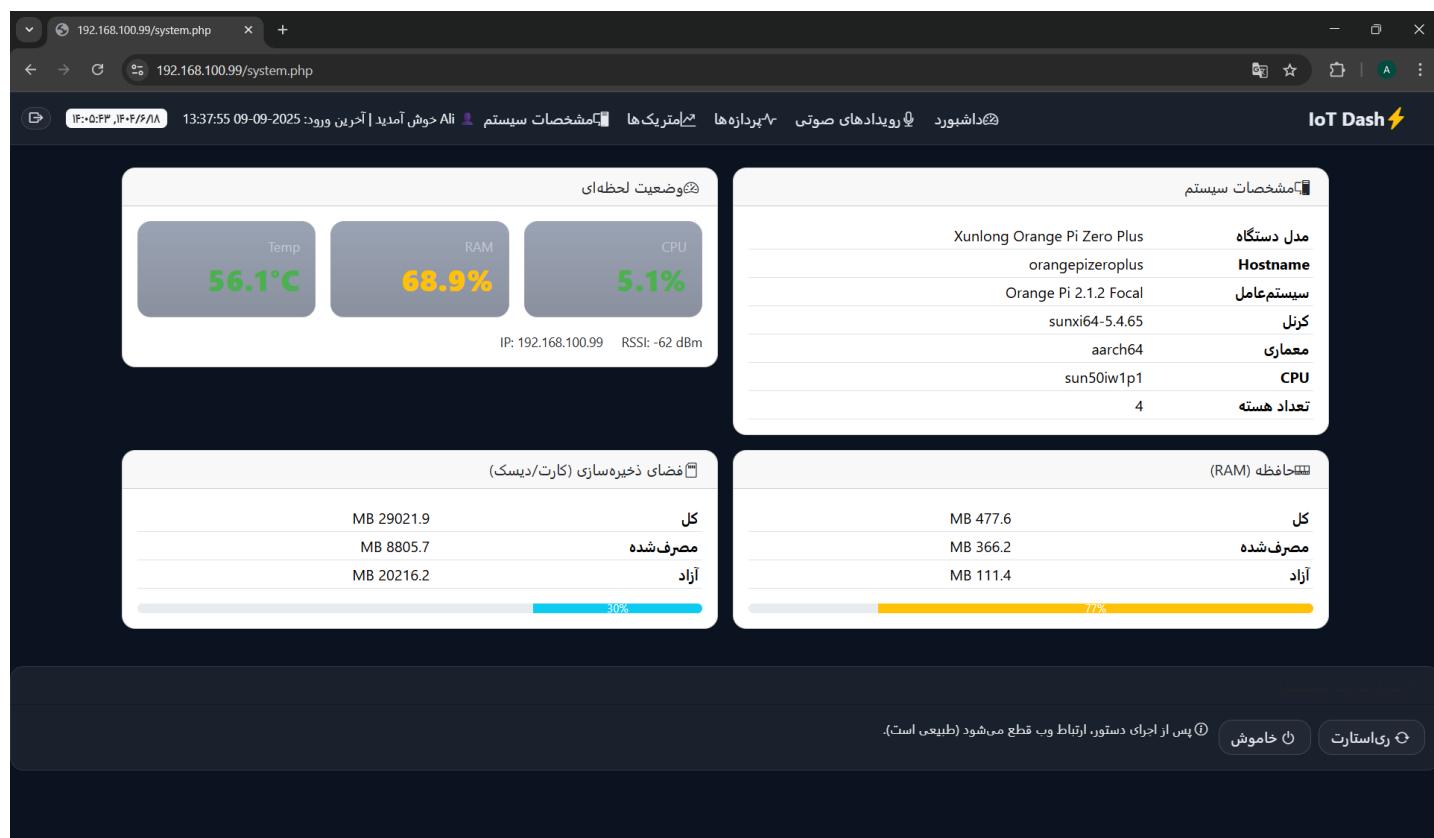
13:29:11 ۰۹-۰۹-۲۰۲۵ | آدرسین ورود | Ali حوش آمدید | آدرسین ورود | امتریک‌ها | پردازه‌ها | مشخصات سیستم | داشبورد | رویدادهای صوتی

IoT Dash

خلاصه وضعیت		نمونه متریک‌ها (آخرین ۲۰ مقدار)					
آلتان	وضعیت:	hog	IP	Temp	%RAM	%CPU	زمان
192.168.100.99	:IP	—	192.168.100.99	54.2	71.3	75	۱۴:۳۶:۵۹ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	53	66.5	2.5	۱۴:۳۶:۵۸ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	53.2	68.8	0	۱۴:۳۶:۵۰ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.1	72.9	10.1	۱۴:۳۶:۴۹ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.6	72.9	6.2	۱۴:۳۶:۴۸ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	51.6	76.1	10.1	۱۴:۳۶:۴۷ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.1	76	6.3	۱۴:۳۶:۴۶ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.1	76	6.3	۱۴:۳۶:۴۵ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	51.6	73.7	15	۱۴:۳۶:۴۴ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	51.6	73.7	12.5	۱۴:۳۶:۴۳ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.3	73.6	5.1	۱۴:۳۶:۴۲ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.3	73.6	5.1	۱۴:۳۶:۴۱ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	55.2	74.1	30.8	۱۴:۳۶:۴۰ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	55.2	74.1	30.8	۱۴:۳۶:۳۹ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	55.2	74.1	32.9	۱۴:۳۶:۳۸ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	55.2	74.1	34.2	۱۴:۳۶:۳۷ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	56.1	76.7	63.2	۱۴:۳۶:۳۶ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	51.1	69.7	2.5	۱۴:۳۶:۳۵ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	52.2	69.7	2.5	۱۴:۳۶:۳۴ ,۱۴*۵/۶/۲۰۲۵
		—	192.168.100.99	51.5	71.3	2.5	۱۴:۳۶:۳۳ ,۱۴*۵/۶/۲۰۲۵

همانطور که مشخص است آخرین مقدار CPU و RAM را همراه با تاریخ و IP سیستم می اندازد که تا 20 مصرف اخیر سیستم چه منابعی و چقدر مصرف شدند همینطور hog منظور است یعنی اون تایم برنامه ای که اجرا شده است منابع زیادی از سیستم استفاده کرده ؟ اگر اینچنین باشد قطعا سیستم آن برنامه را غیر فعال به اصلاح Kill میکند اگر این اتفاق بیوفتد به جای خط جلویش تیک میوافت (ما سیاست را اینگونه تعريف کردیم که اگر 5 ثانیه CPU یا RAM 99٪ استفاده شدند آن برنامه ای که باعث این اتفاق شده است به طور خودکار بسته به اصلاح Kill شود).

صفحه مشخصات سیستم :



صفحه تنظیمات تشخیص صدا :

192.168.100.99/audio_settings.php

داشبورد رویدادهای صوتی پردازهای خوش آمدید | آخرین ورود: ۰۶-۰۹-۲۰۲۵ ۲۳:۴۹:۰۴

IoT Dash

تنظیمات تشخیص صدا

حالت تشخیص

استفاده از مدل ML (برای دقق بالاتر توصیه می‌شود، اما منابع بیشتری مصرف می‌کند) زمانی از استفاده کنید که نیاز به دقق بیشتر دارید؛ حالت بدون ML سینکتلر و سریع‌تر است.

زمان نمونه‌برداری (ثانیه)

17 16 15 14 13 12 11 10 9 8 7 6 5
 20 19 18

این مقدار، طول «گوش‌دادن» در هر بار نمونه‌برداری را تعیین می‌کند.

[ذخیره تنظیمات] بازگشت به داشبورد

• حالت: ساده • زمان نمونه‌برداری: 10 ثانیه

AUDIO_CFG_FILE

192.168.100.99/audio_settings.php

داشبورد رویدادهای صوتی پردازهای خوش آمدید | آخرین ورود: ۰۹-۰۹-۲۰۲۵ ۱۳:۳۷:۵۵

IoT Dash

تنظیمات تشخیص صدا

تنظیمات با موفقیت ذخیره شد.

حالت تشخیص

استفاده از مدل ML (برای دقق بالاتر توصیه می‌شود، اما منابع بیشتری مصرف می‌کند) زمانی از استفاده کنید که نیاز به دقق بیشتر دارید؛ حالت بدون ML سینکتلر و سریع‌تر است.

زمان نمونه‌برداری (ثانیه)

17 16 15 14 13 12 11 10 9 8 7 6 5
 20 19 18

این مقدار، طول «گوش‌دادن» در هر بار نمونه‌برداری را تعیین می‌کند.

[ذخیره تنظیمات] بازگشت به داشبورد

• حالت: ML • زمان نمونه‌برداری: 12 ثانیه

AUDIO_CFG_FILE

صفحه محدودسازی رویدادها در تشخیص صدا :

192.168.100.99/audio_blocklist.php

داشبورد رویدادهای صوتی ۷۰٪ مشخصات سیستم Ali خوش آمدید | آخرین ورود: 12:03:42 11-09-2025

IoT Dash

محدودسازی رویدادها در تشخیص صدا

موارد تک-خودرده در تشخیص صوت نادیده گرفته می‌شوند.
فایل CSV کلاس‌ها از assets/yamnet_class_map.csv خوانده می‌شود.

جستجوی نام رویداد...

انتخاب همه عدم انتخاب معکوس

ذخیره تغییرات

برگشت به داشبورد

- A capella
- Accelerating, revving, vroom
- Accordion
- Acoustic guitar
- Afrobeat
- Air brake
- Air conditioning
- Aircraft
- Aircraft engine
- Air horn, truck horn
- Alarm
- Alarm clock
- Ambient music
- Ambulance (siren)
- Angry music

192.168.100.99/audio_blocklist.php

داشبورد رویدادهای صوتی ۷۰٪ مشخصات سیستم Ali خوش آمدید | آخرین ورود: 12:03:42 11-09-2025

IoT Dash

محدودسازی رویدادها در تشخیص صدا

موارد تک-خودرده در تشخیص صوت نادیده گرفته می‌شوند.
فایل CSV کلاس‌ها از assets/yamnet_class_map.csv خوانده می‌شود.

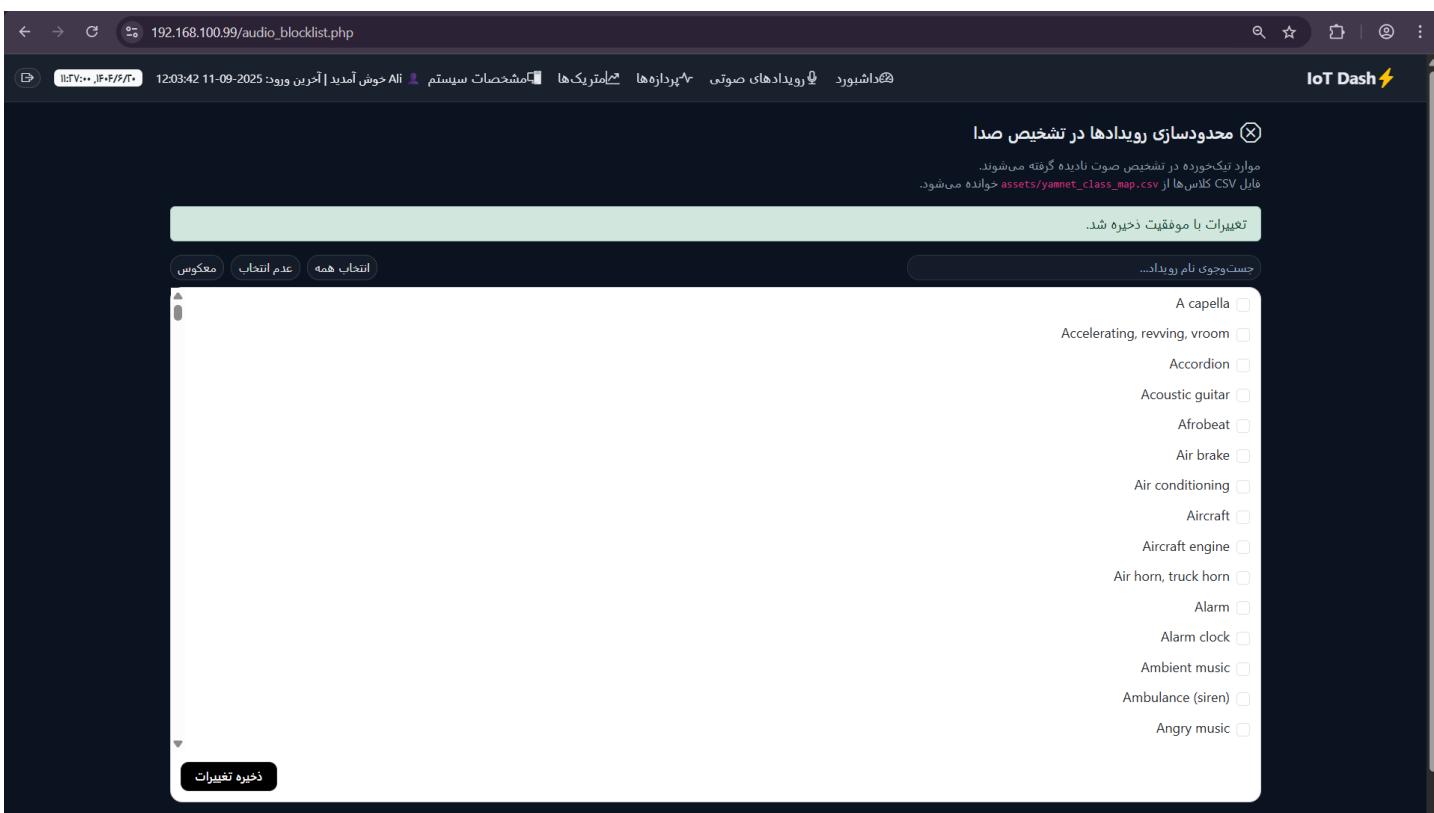
bell

انتخاب همه عدم انتخاب معکوس

ذخیره تغییرات

برگشت به داشبورد

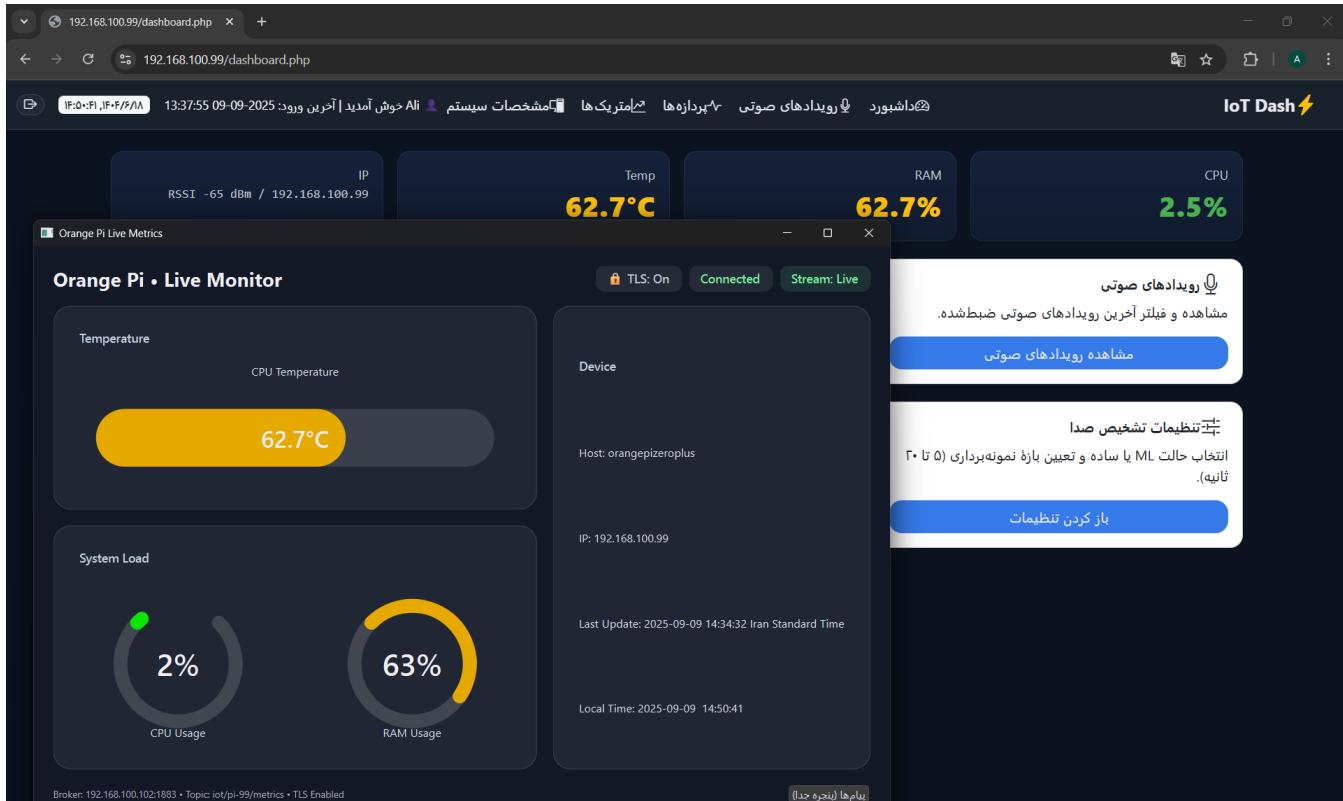
- Bell
- Bellow
- Belly laugh
- Bicycle bell
- Church bell
- Cowbell
- Doorbell
- Jingle bell
- Telephone bell ringing
- Tubular bells



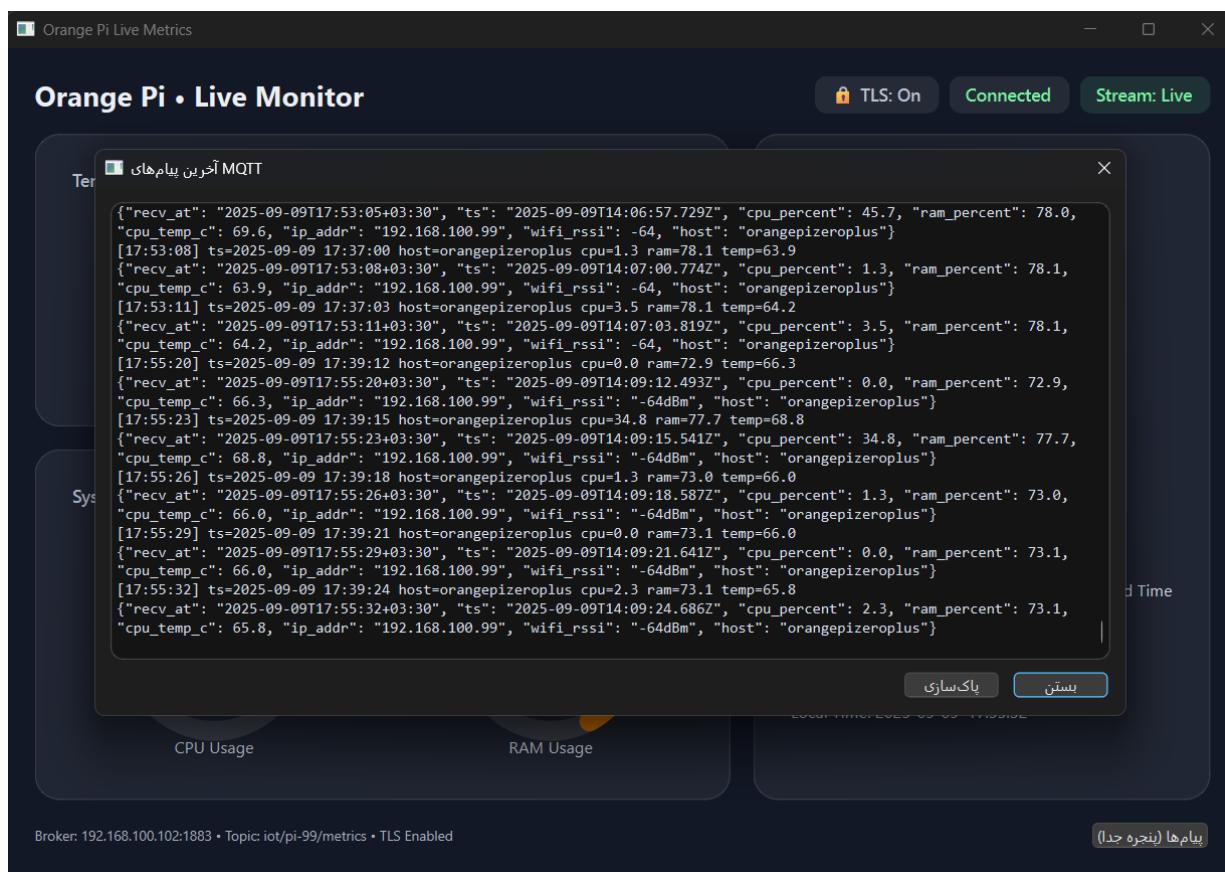
بعد ما سیاست ریست سیستم داریم `systemd service with auto-restart and resource limits` چنانچه CPU یا RAM بیشتر از 5 ثانیه 99٪ بشوند اول برنامه سعی میکند که برنامه ای که بیشترین مصرف را از RAM یا CPU دارد بند چانچه برنامه بسته نشود یا سطح دسترسی برایش تعريف شده باشد که غیر قابل Kill کردن باشد سیستم خودش ریست میشود برنامه `stress.py` را 99٪ میکند ما این برنامه را جوری تعريف کردیم توسط `systemd` که به هیچ عنوان توسط سیستم قابلیت Kill نداشته باشد برای همین سیستم بعد از 5 ثانیه خودکار ریست میشود ولی برنامه `burn_ram.py` به صورت سیستمی قفل نشده است برای همین بعد از 5 ثانیه خودکار سیستم خودش برنامه را Kill میکند

```
ali@orangepizeroplus:/opt/iot/python$ python3 burn_ram.py
Terminated
ali@orangepizeroplus:/opt/iot/python$
```

حال برویم سراغ MQTT و بروکر Mosquitto با ACL و TLS طراحی شده برای ساپسکرایب کردن: تطبیق دما و میزان استفاده پردازنده و رم با وب سایت



پیام ها شامل timestamp استاندارد (8610) هستند



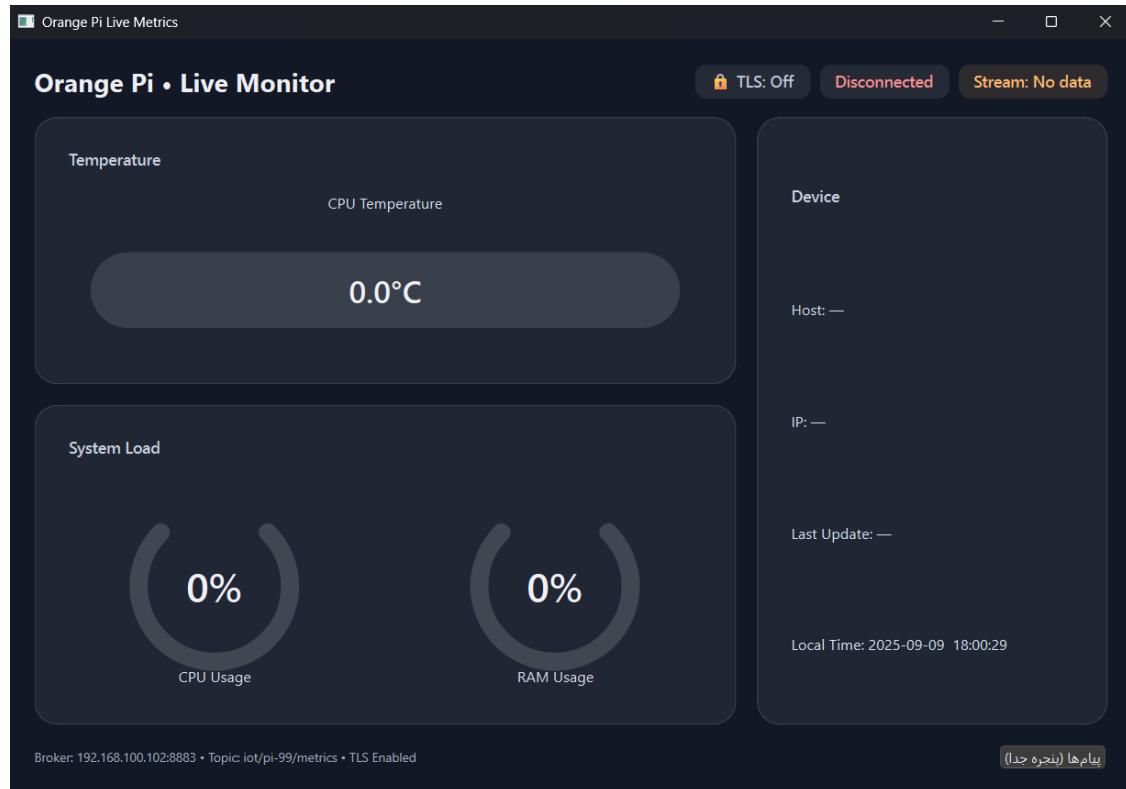
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF

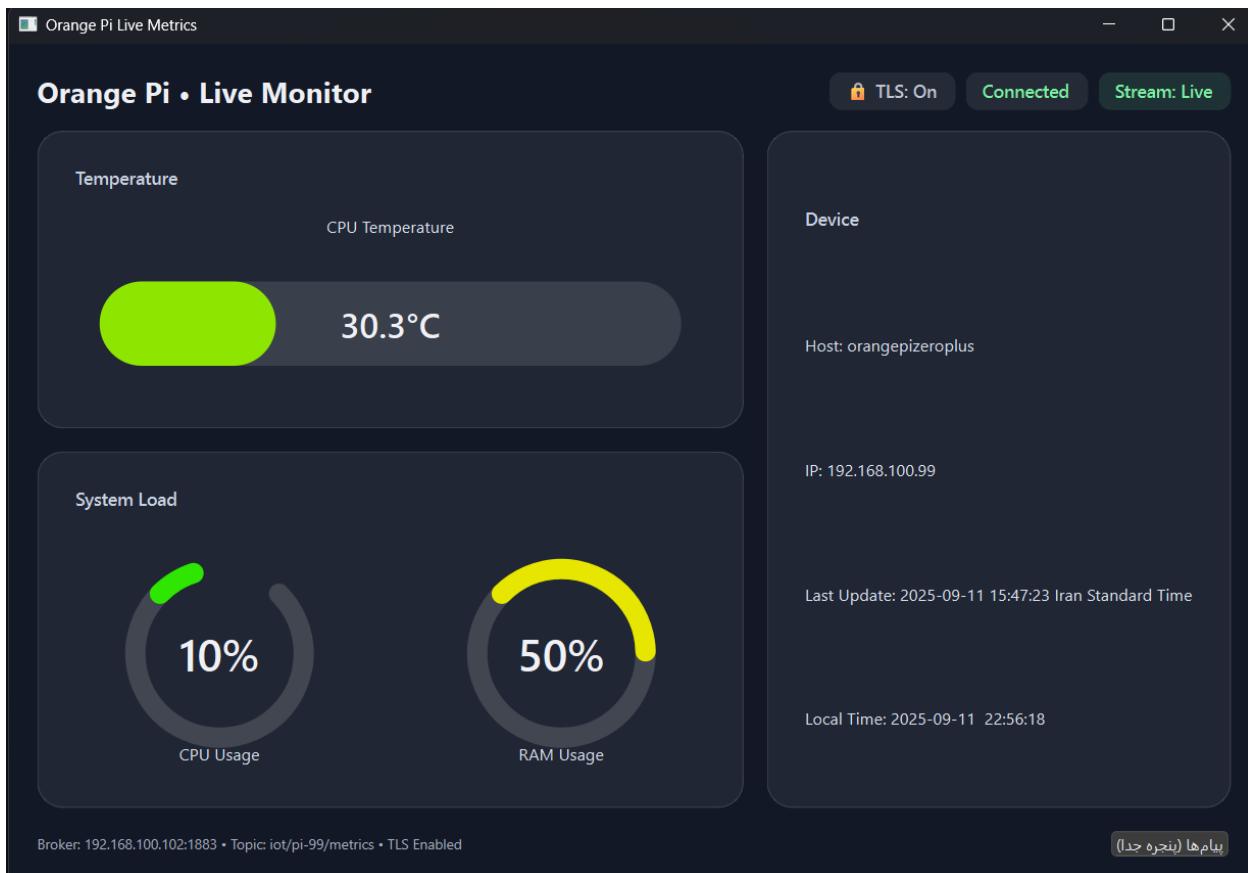
[{"recv_at": "2025-09-09T17:55:57+03:30", "ts": "2025-09-09T14:09:49.065Z", "cpu_percent": 0.0, "ram_percent": 73.2, "cpu_temp_c": 65.2, "ip_addr": "192.168.100.99", "wifi_rssi": "-64dBm", "host": "orangepirzeroplus"} [17:56:00] ts=2025-09-09 17:39:52 host=orangepirzeroplus cpu=2.3 ram=75.5 temp=67.5 {"recv_at": "2025-09-09T17:56:00+03:30", "ts": "2025-09-09T14:09:52.097Z", "cpu_percent": 2.3, "ram_percent": 75.5, "cpu_temp_c": 67.5, "ip_addr": "192.168.100.99", "wifi_rssi": "-64dBm", "host": "orangepirzeroplus"} [17:56:03] ts=2025-09-09 17:39:55 host=orangepirzeroplus cpu=0.5 ram=75.4 temp=66.3 {"recv_at": "2025-09-09T17:56:03+03:30", "ts": "2025-09-09T14:09:55.145Z", "cpu_percent": 0.5, "ram_percent": 75.4, "cpu_temp_c": 66.3, "ip_addr": "192.168.100.99", "wifi_rssi": "-64dBm", "host": "orangepirzeroplus"} [17:56:06] ts=2025-09-09 17:39:58 host=orangepirzeroplus cpu=3.5 ram=75.6 temp=66.3 {"recv_at": "2025-09-09T17:56:06+03:30", "ts": "2025-09-09T14:09:58.202Z", "cpu_percent": 3.5, "ram_percent": 75.6, "cpu_temp_c": 66.3, "ip_addr": "192.168.100.99", "wifi_rssi": "-64dBm", "host": "orangepirzeroplus"} [17:56:09] ts=2025-09-09 17:40:01 host=orangepirzeroplus cpu=7.7 ram=75.6 temp=66.3 {"recv_at": "2025-09-09T17:56:09+03:30", "ts": "2025-09-09T14:10:01.253Z", "cpu_percent": 7.7, "ram_percent": 75.6, "cpu_temp_c": 66.3, "ip_addr": "192.168.100.99", "wifi_rssi": "-64dBm", "host": "orangepirzeroplus"}

```

حال اگر ارتباط با Orange Pi قطع شود:



اگر ارتباط با Orange Pi قطع نشود ولی دیتایی هم شامل میزان مصرف رم و پردازنده و دما از Orange Pi به تایپیک پابلیشن نشود:



: LWT برای

توی کدمان گذاشتیم $\text{keepalive} = 30$

یعنی اگر کلاینت تا 30 ثانیه هیچ پیامی/پینگ نفرسته، بروکر می‌فهمه مشکوک شده.

استاندارد MQTT می‌گه بعد از $1.5 \times \text{keepalive}$ کانکشن رو مرده حساب کن.

پس حدود ۴۵ ثانیه بعد، بروکر کانکشن رو می‌بنده و همون لحظه پیام (offline) LWT (offline) را توی تاپیک `iot/pi-99/lwt` منتشر می‌کنه.

در نتیجه

برد در کد `publish` می‌کنه وقتی وصل می‌شیه.

offline

اگر تمیز `disconnect` کنیم → برد در تاپیک `publish` می‌کنه

اگر ناگهانی قطع بشه → بروکر به جای برد `publish` می‌کنه این همون LWT هست

وقتی کلاینت قطع می شه و هیچ پیامی نمی ده، خود بروکر **Mosquitto** رو در تاپیک **LWT** بعد از ~45 ثانیه پیام **offline** فرسته، چون ما هنگام اتصال اون پیام رو به عنوان **LWT** ثبت کردیم.

دیدن تاپیک **LWT** با بروکر **Mosquitto** از طریق **CMD** یا پاور شل ویندوز :

```
\mosquitto_sub.exe -h 192.168.100.102 -p 8883 --cafile " C:\mosquitto_certs\server.crt
```

```
" -u Ali -P EmbeddedProject -t "iot/pi-99/lwt" -v
```

```
Windows PowerShell
iot/pi-99/lwt {"status": "online", "src": "client", "ts": "2025-09-09T17:17:16.819Z", "host": "orangepireoplus"}
iot/pi-99/lwt {"status": "offline", "src": "will", "ts": "2025-09-09T17:17:13.688Z", "host": "orangepireoplus"}
iot/pi-99/lwt {"status": "online", "src": "client", "ts": "2025-09-09T17:52:05.712Z", "host": "orangepireoplus"}|
```

ساعتهایی که توی پیام‌های MQTT می‌بینیم ("ts": "2025-09-06T10:03:38.785Z") فرمات استاندارد زمان ISO 8601 هست.

2025-09-06 تاریخ (۶ سپتامبر)

T جداولنده تاریخ و زمان

10:03:38.785 ساعت و ۷۸۵ میلی ثانیه

Z یعنی این زمان به UTC زمان جهانی ثبت شده

اگه Orange Pi روش بشه و آنلاین بشه به صورت خودکار پیام آنلاین در تاپیک **lwt** ارسال میکنه و جوری تنظیم کردم که مشخص بشه پیام از **client src** هست

اگه به حالت معمولی برنامه رو در Orange Pi ببندیم خود **lwt** به تاپیک **Orange Pi** میگه من رفتم یعنی پیام **offline** میفرسته پس **src** باز هم **client** هست ولی اگه بیهوی قطع بشه مثلا برقصش بره هیچی نمیگه دیگه طبق تنظیماتی که انجام دادیم بروکر خودش 45 ثانیه صبر میکنه اگه خبری ازش نشد پیام **offline** رو میفرسته و **src will** میشه

```
iot/pi-99/lwt {"status": "online", "src": "client", "ts": "2025-09-09T17:17:16.819Z", "host": "orangepireoplus"}
```

این پیام رو خود پابلیشر (اسکریپت) بعد از وصل شدن موفق به بروکر منتشر کرده.

src client یعنی مشخصه این پیام توسط کلاینت فرستاده شده (نه بروکر).

ts UTC تایم استمپ

(orangepirerooplus) اسم دستگاه **host**

```
iot/pi-99/lwt {"status": "offline", "src": "will", "ts": "2025-09-09T17:17:13.688Z", "host": "orangepirerooplus"}
```

این پیام رو نه اسکریپت، بلکه خود بروکر منتشر کرده.

معنی پیام **srcwill** LWT (Last Will & Testament) بوده.

معنی ارتباط کلاینت قبلی درست بسته نشده مثلاً کرش کرده، یا با **client_id** یکسان دوباره وصل شده و بروکر مجبور شده پیام "offline" را طبق وصیت‌نامه‌ی کلاینت پخش کند.

```
iot/pi-99/lwt {"status": "online", "src": "client", "ts": "2025-09-09T17:52:05.712Z", "host": "orangepirerooplus"}
```

دوباره **online** از طرف کلاینت. یعنی اسکریپت یا سرویس بعد از قطع/ریستارت دوباره وصل شده و خودش پیام آنلاین داده.

وقتی **src = "client"** پیام مستقیماً توسط برنامه‌ی Orange Pi منتشر شده (وصل یا خروج تمیز).

وقتی **src = "will"** پیام توسط بروکر منتشر شده چون کلاینت تمیز دیسکانکت نکرده کرش، 9-kill، قطعی برق، یا اتصال جدید با همان **client_id**.

Robustness: error handling, meaningful logs, no crashes on network همینطور

disconnects داریم و تمامی مشکلات از طریق json بر میگردند و قابلیت log گیری از سایت داریم

برای دیدن log ها و json ها از سایت :

https://192.168.100.99/iotdash/api/events_list.php?per=5

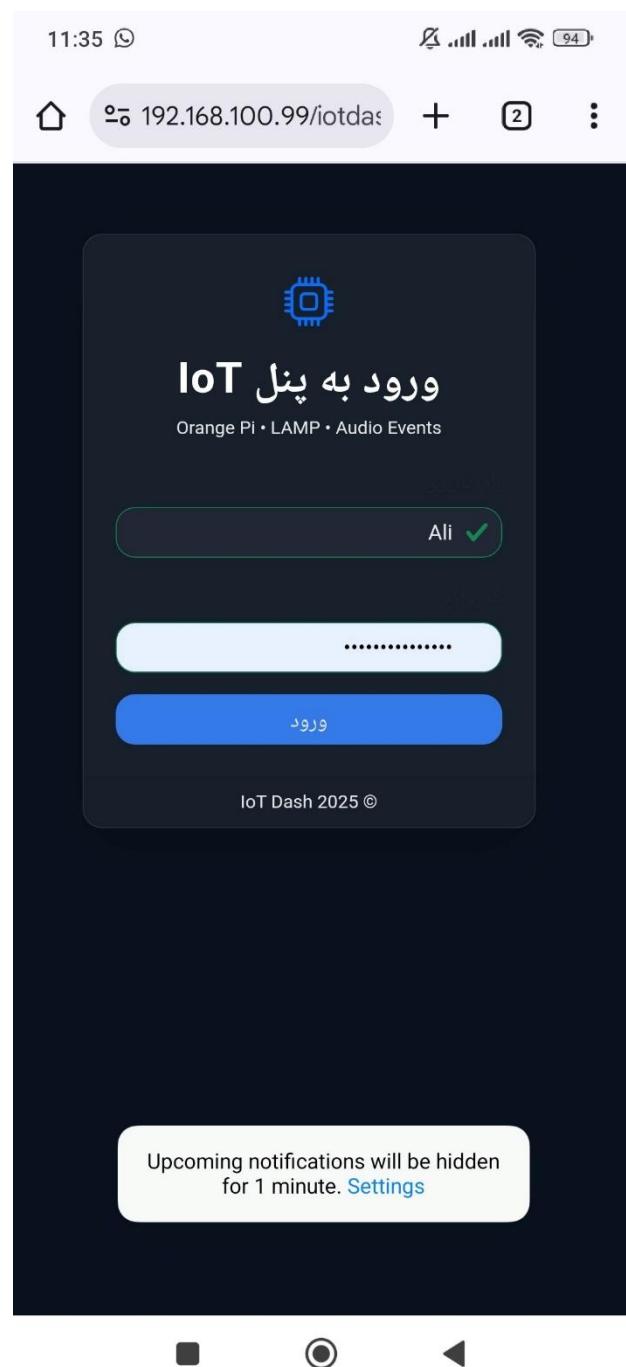
```
Pretty-print   
  
{  
    "rows": [  
        {  
            "id": 236,  
            "ts": "2025-09-05 18:00:17",  
            "label": "Silence",  
            "confidence": "78.00",  
            "node_id": "Ali-Node"  
        },  
        {  
            "id": 235,  
            "ts": "2025-09-05 18:00:05",  
            "label": "Silence",  
            "confidence": "84.00",  
            "node_id": "Ali-Node"  
        },  
        {  
            "id": 234,  
            "ts": "2025-09-05 15:10:30",  
            "label": "Silence",  
            "confidence": "91.00",  
            "node_id": "Ali-Node"  
        },  
        {  
            "id": 233,  
            "ts": "2025-09-05 15:10:18",  
            "label": "Silence",  
            "confidence": "94.00",  
            "node_id": "Ali-Node"  
        },  
        {  
            "id": 232,  
            "ts": "2025-09-04 15:25:20",  
            "label": "Silence",  
            "confidence": "87.00",  
            "node_id": "Ali-Node"  
        }  
    ],  
    "page": 1,  
    "per": 5,  
    "total": 236,  
    "pages": 48  
}
```

Docker هم برای پروژه ساختیم

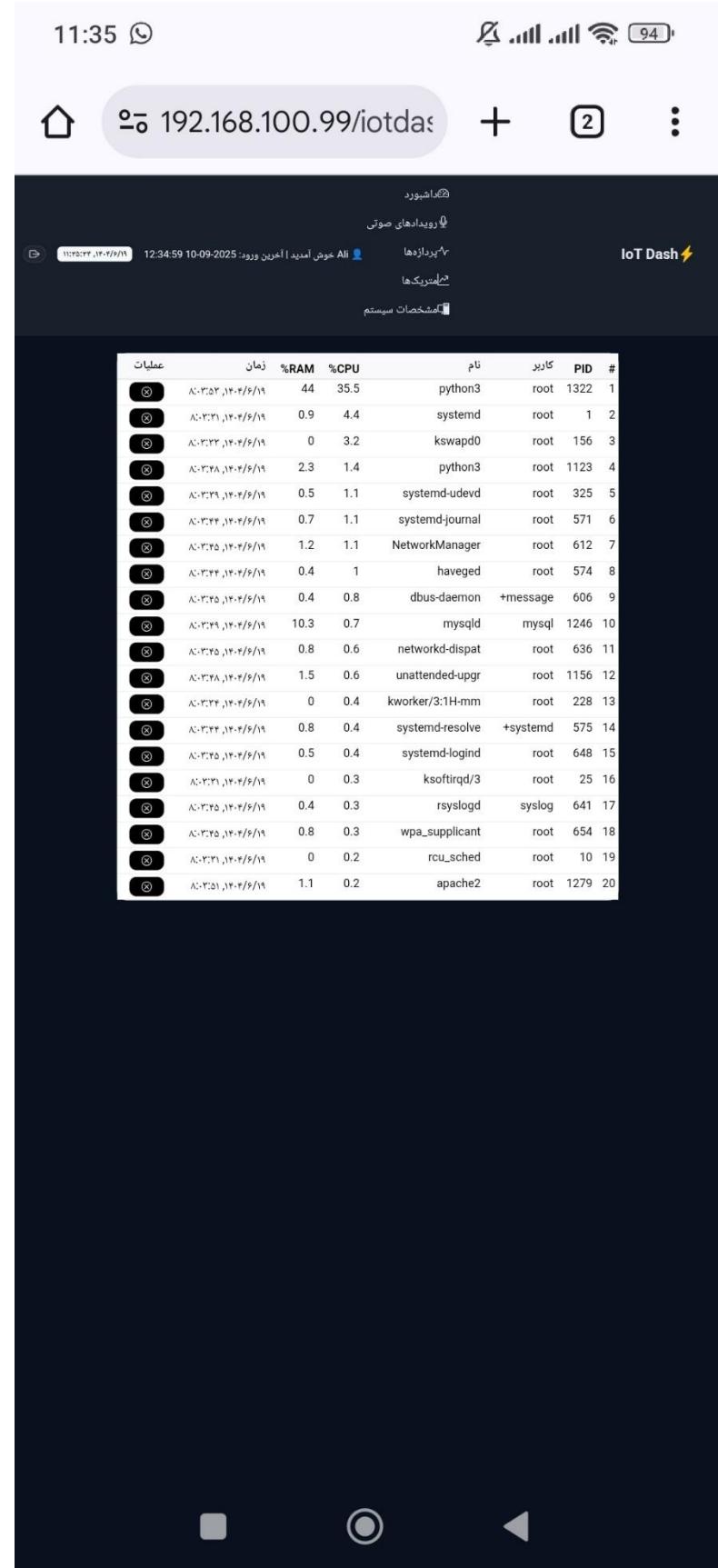
محیط ایزوله برای وب سرور

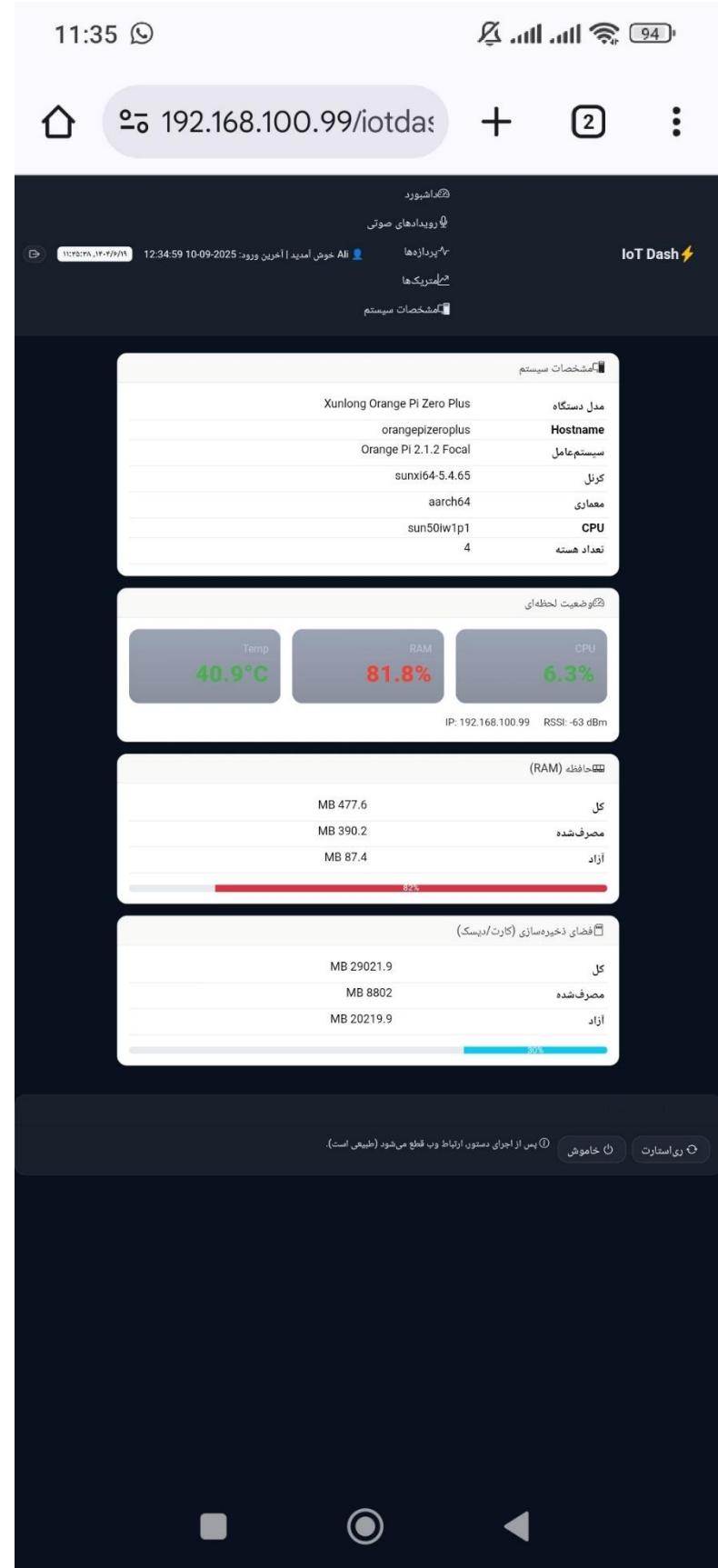
به جای نصب Apache/Nginx/PHP/MySQL روی Orange Pi ، می توانی همه روتی کانتینرها بالا بیاریم. با کمک Docker مثلا یک کانتینر برای Apache/PHP (dashboard) یک کانتینر برای MariaDB یک کانتینر برای MQTT broker (Mosquitto)

سازگاری وب با موبایل :

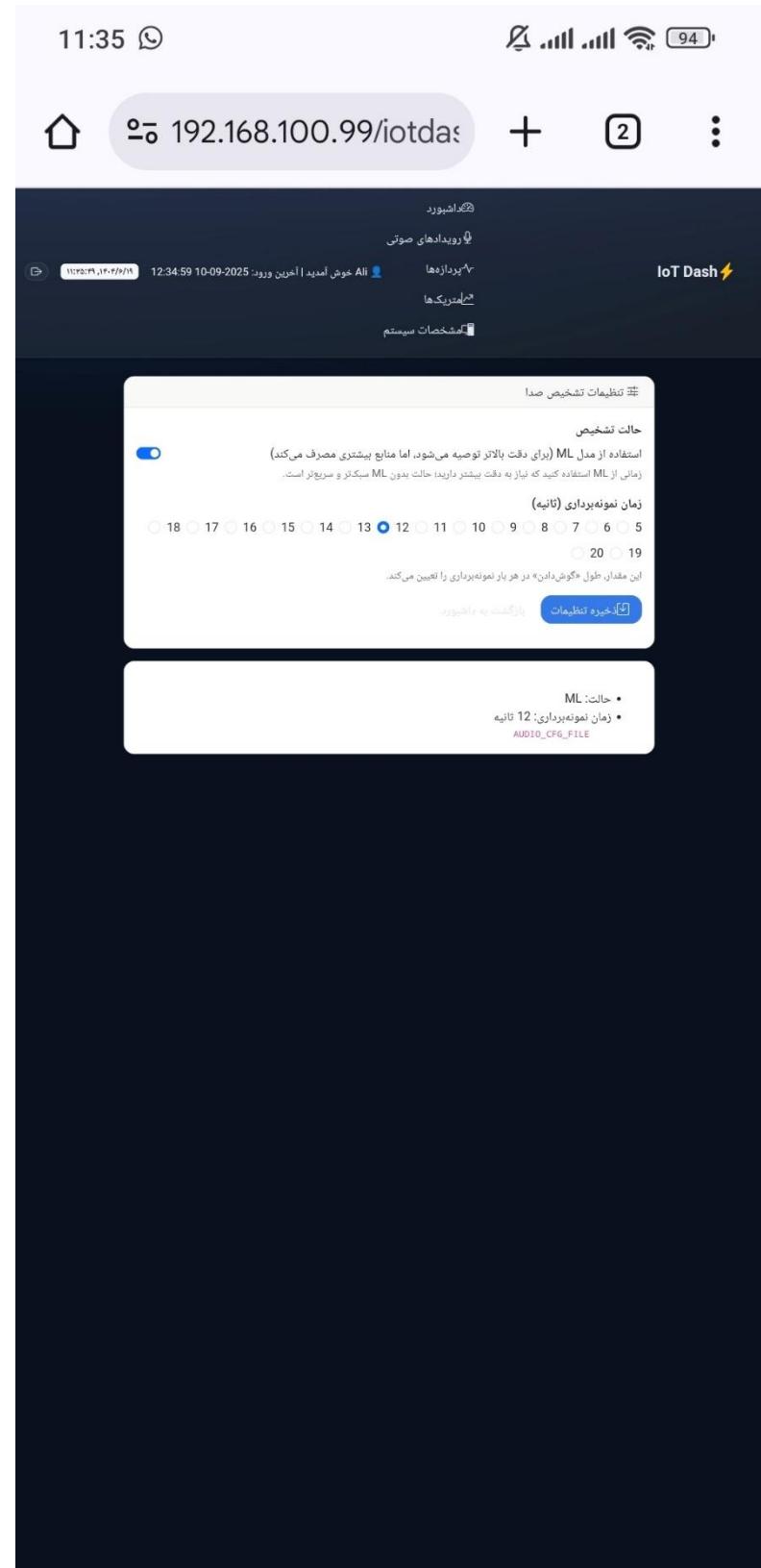












2- نکات مهم در روند پروژه:

رونند کار اگه از 0 تا 100 گفته شود احتمالا بالای 300 صفحه میشود من سعی میکنم که نکات مهم کار رو باز گو کنم.

اول باید کلی کتابخانه و برنامه روی Orange Pi نصب کنیم از پایتون بگیر تا بروکر ماسکیتو و دینتا بیس و ... php

```
root@orangepizeroplus:~# apt install python3-pip -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.11).
0 upgraded, 0 newly installed, 0 to remove and 239 not upgraded.
root@orangepizeroplus:~# pip3 install tensorflow sounddevice numpy
Collecting tensorflow
  Downloading tensorflow-2.13.1-cp38-cp38-manylinux_2_17_aarch64.manylinux2014_a
  arch64.whl (2.0 kB)
Collecting sounddevice
  Downloading sounddevice-0.5.2-py3-none-any.whl (32 kB)
Collecting numpy
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_aarch64.manylinux2014_aarch6
  4.whl (14.0 MB)
    |████████████████████████████████| 14.0 MB 10.2 MB/s eta 0:00:01
    |████████████████████████████████| 10.4 MB 10.2 MB/s eta 0:00:01 eta 0:00:00
    |████████████████████████████████| 12.7 MB 10.2 MB/s eta 0:00:01 eta 0:00:00
    |████████████████████████████████| 13.3 MB 10.2 MB/s eta 0:00:013.3 MB 10.2
    |████████████████████████████████| 14.0 MB 2.9 kB/s
Collecting tensorflow-cpu-aws==2.13.1; platform_system == "Linux" and (platform_
machine == "arm64" or platform_machine == "aarch64")
  Downloading tensorflow_cpu_aws-2.13.1-cp38-cp38-manylinux_2_17_aarch64.manylin
  ux2014_aarch64.whl (248.7 MB)
    |████████████████████████████████| 72.9 MB 10.6 MB/s
```

```
Collecting sounddevice
  Downloading sounddevice-0.5.2-py3-none-any.whl (32 kB)
Collecting CFFI>=1.0
  Downloading cffi-1.17.1-cp38-cp38-manylinux_2_17_aarch64.manylinux2014_aarch64
  .whl (448 kB)
    |████████████████████████████████| 448 kB 887 kB/s
Collecting pycparser
  Downloading pycparser-2.22-py3-none-any.whl (117 kB)
    |████████████████████████████████| 117 kB 11.9 MB/s
Installing collected packages: pycparser, CFFI, sounddevice
Successfully installed CFFI-1.17.1 pycparser-2.22 sounddevice-0.5.2
root@orangepizeroplus:~#
root@orangepizeroplus:~#
```

```
Collecting pycparser
  Downloading pycparser-2.22-py3-none-any.whl (117 kB)
    |████████████████████████████████| 117 kB 11.9 MB/s
Installing collected packages: pycparser, CFFI, sounddevice
Successfully installed CFFI-1.17.1 pycparser-2.22 sounddevice-0.5.2
root@orangepizeroplus:~#
root@orangepizeroplus:~# apt install -y python3-numpy
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```

python3-numpy is already the newest version (1:1.17.4-5ubuntu3.1).
0 upgraded, 0 newly installed, 0 to remove and 221 not upgraded.
root@orangepizeroplus:~# TMPDIR=/root/tmp pip3 install --no-cache-dir --upgrade
pip
TMPDIR=/root/tmp pip3 install --no-cache-dir sounddevice
TMPDIR=/root/tmp pip3 install --no-cache-dir "grpcio>=1.54" --only-binary=:all:
TMPDIR=/root/tmp pip3 install --no-cache-dir tensorflow-aarch64==2.13.0
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
    |██████████| 1.8 MB 628 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2

```

```

root@orangepizeroplus:~# python3 -c <<'PY'
> import numpy, h5py
> print("numpy:", numpy.__version__)
> print("h5py:", h5py.__version__)
> import tensorflow as tf
> print("tensorflow:", tf.__version__)
> PY
numpy: 1.24.3
h5py: 3.10.0
tensorflow: 2.13.0
root@orangepizeroplus:~#

```

یک swap 2 گیگی هم میسازیم تا اگر رم ما کم اوmd از swap استفاده کند برای ایجاد swap از فضای sd card میکند

```

root@orangepizeroplus:~# mkdir -p /root/tmp
root@orangepizeroplus:~# fallocate -l 2G /swapfile || dd if=/dev/zero of=/swapfile bs=1M count=2048
root@orangepizeroplus:~# chmod 600 /swapfile
root@orangepizeroplus:~# mkswap /swapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=1d232e6f-f876-4a18-8dab-05f9f8d9a72e
root@orangepizeroplus:~# swapon /swapfile
root@orangepizeroplus:~# # دائمیاً کن:
root@orangepizeroplus:~# grep -q "/swapfile" /etc/fstab || echo "/swapfile none
swap sw 0 0" >> /etc/fstab
root@orangepizeroplus:~#

```

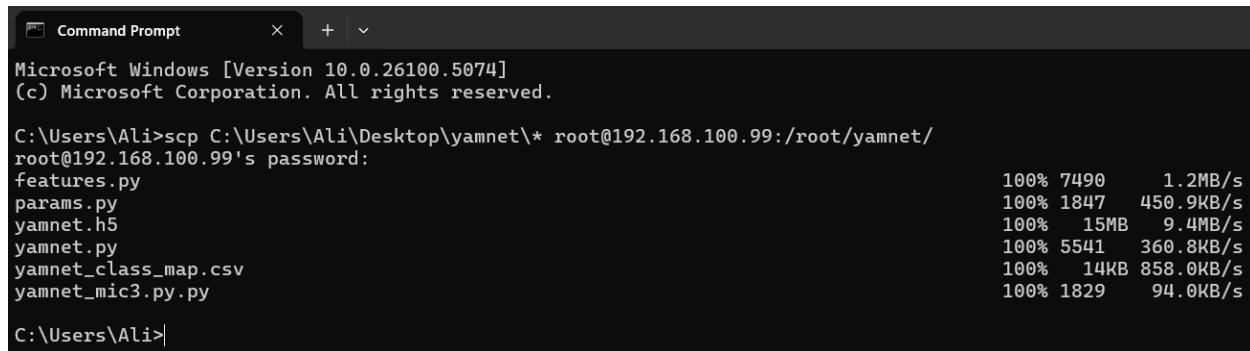
NAME	TYPE	SIZE	USED	PRIORITY
/swapfile	file	2G	10.5M	-2
/dev/zram1	partition	238.8M	30.5M	5

```

root@orangepizeroplus:~# free -h
      total        used         free        shared   buff/cache   available
Mem:       477Mi       86Mi      206Mi        1.0Mi      184Mi      378Mi
Swap:      2.2Gi      40Mi      2.2Gi
root@orangepizeroplus:~#

```

فایل هایی که برای پردازش صدا روی لینوکس ساختیم را با این روش ساده به Orange Pi منتقل میکنیم:



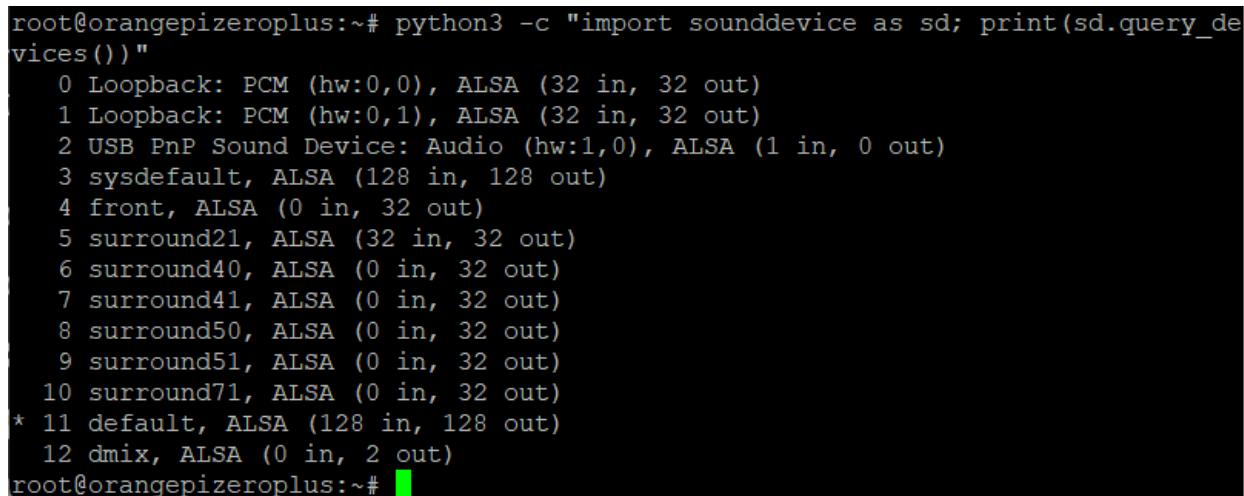
```
Command Prompt x + v
Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ali>scp C:\Users\Ali\Desktop\yamnet\* root@192.168.100.99:/root/yamnet/
root@192.168.100.99's password:
features.py                                100% 7490      1.2MB/s
params.py                                  100% 1847     450.9KB/s
yamnet.h5                                   100%   15MB     9.4MB/s
yamnet.py                                   100% 5541     360.8KB/s
yamnet_class_map.csv                      100% 14KB    858.0KB/s
yamnet_mic3.py.py                         100% 1829     94.0KB/s

C:\Users\Ali>
```

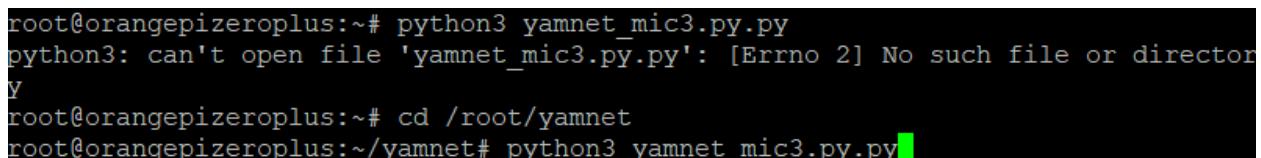
پس از اتصال میکروفون با usb به Orange Pi میتوانیم با دستور زیر بفهمیم که به درستی به دستگاه متصل شده است شماره

دوم

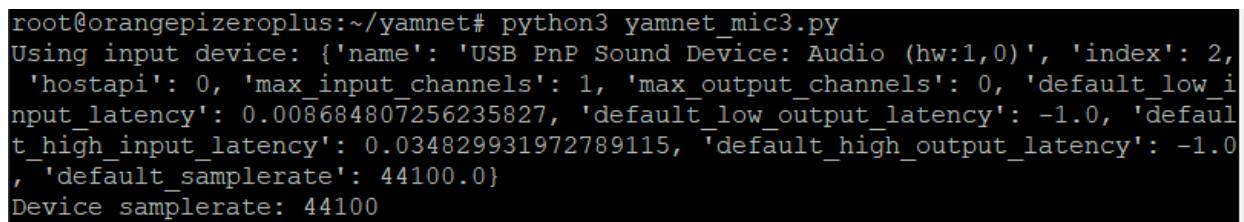


```
root@orangeplus:~# python3 -c "import sounddevice as sd; print(sd.query_devices())"
0 Loopback: PCM (hw:0,0), ALSA (32 in, 32 out)
1 Loopback: PCM (hw:0,1), ALSA (32 in, 32 out)
2 USB PnP Sound Device: Audio (hw:1,0), ALSA (1 in, 0 out)
3 sysdefault, ALSA (128 in, 128 out)
4 front, ALSA (0 in, 32 out)
5 surround21, ALSA (32 in, 32 out)
6 surround40, ALSA (0 in, 32 out)
7 surround41, ALSA (0 in, 32 out)
8 surround50, ALSA (0 in, 32 out)
9 surround51, ALSA (0 in, 32 out)
10 surround71, ALSA (0 in, 32 out)
* 11 default, ALSA (128 in, 128 out)
12 dmix, ALSA (0 in, 2 out)
root@orangeplus:~#
```

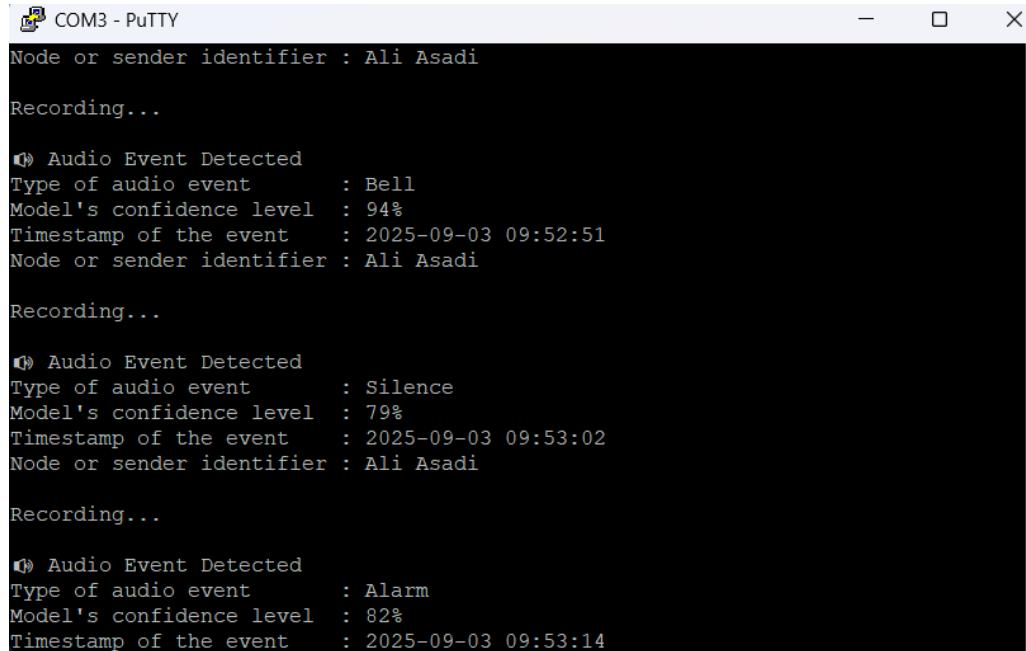
حال میتوانیم برنامه مان را (برنامه اولیه پردازش صدا) را اجرا کنیم و ببینیم که به درستی روی Orange Pi کار میکند و دستگاه صدای این را تشخیص میدهد.



```
root@orangeplus:~# python3 yamnet_mic3.py.py
python3: can't open file 'yamnet_mic3.py.py': [Errno 2] No such file or directory
root@orangeplus:~# cd /root/yamnet
root@orangeplus:~/yamnet# python3 yamnet_mic3.py.py
```



```
root@orangeplus:~/yamnet# python3 yamnet_mic3.py
Using input device: {'name': 'USB PnP Sound Device: Audio (hw:1,0)', 'index': 2,
 'hostapi': 0, 'max_input_channels': 1, 'max_output_channels': 0, 'default_low_input_latency': 0.008684807256235827, 'default_low_output_latency': -1.0, 'default_high_input_latency': 0.034829931972789115, 'default_high_output_latency': -1.0, 'default_samplerate': 44100.0}
Device samplerate: 44100
```



```
COM3 - PuTTY
Node or sender identifier : Ali Asadi
Recording...
    ☺ Audio Event Detected
    Type of audio event      : Bell
    Model's confidence level : 94%
    Timestamp of the event   : 2025-09-03 09:52:51
    Node or sender identifier : Ali Asadi
Recording...
    ☺ Audio Event Detected
    Type of audio event      : Silence
    Model's confidence level : 79%
    Timestamp of the event   : 2025-09-03 09:53:02
    Node or sender identifier : Ali Asadi
Recording...
    ☺ Audio Event Detected
    Type of audio event      : Alarm
    Model's confidence level : 82%
    Timestamp of the event   : 2025-09-03 09:53:14
```

برای دیتا بیس :

```
root@orangepizeroplus:~# sudo mysql
sudo: mysql: command not found
root@orangepizeroplus:~# sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease
Reading package lists... 99%

Your MariaDB connection id is 36
Server version: 10.3.39-MariaDB-0ubuntu0.20.04.2 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE IF NOT EXISTS sound_events CHARACTER SET utf8mb4;
Query OK, 1 row affected (0.019 sec)

MariaDB [(none)]> CREATE USER IF NOT EXISTS 'Ali'@'localhost' IDENTIFIED BY 'EmbeddedProject';
Query OK, 0 rows affected (0.002 sec)

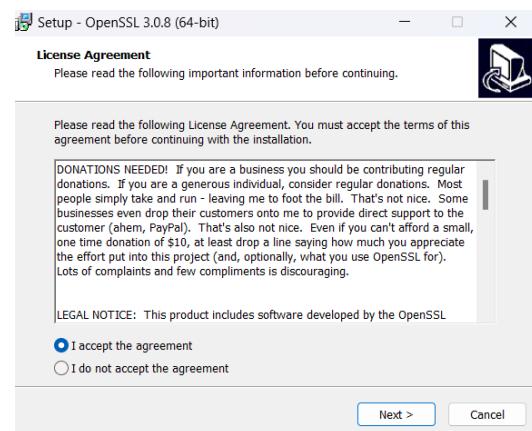
MariaDB [(none)]> GRANT SELECT, INSERT ON sound_events.* TO 'Ali'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> EXIT;
Bye
root@orangepizeroplus:~#
```

برای ایمن کردن با TLS بروکر ماسکیتو باید این کار هارو انجام بدیم:

برنامه open ssl را روی ویندوز نصب میکنیم.



هنگام نصب، گزینه "The OpenSSL binaries (/bin) directory" را برای اضافه کردن به مسیر PATH انتخاب کنیم.

C:\Program Files\OpenSSL\bin

برای اطمینان از اینکه OpenSSL به درستی نصب شده است، دستور زیر را در CMD وارد کنیم:

openssl version

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ali>openssl version
OpenSSL 3.0.8 7 Feb 2023 (Library: OpenSSL 3.0.8 7 Feb 2023)

C:\Users\Ali>
```

ویندوز open sslCmd را شناخت.

در فایل کانفینگ این موارد رو ادیت یا اضافه میکنیم :

listener 8883

cafile C:\mosquitto_certs\server.crt

certfile C:\mosquitto_certs\server.crt

keyfile C:\mosquitto_certs\server.key

protocol mqtt

تعریف فایل رمز عبور

allow_anonymous false

password_file C:\mosquitto\mosquitto_passwd_file

acl_file C:\mosquitto\aclfile

دستورهای زیر را برای ایجاد گواهینامه استفاده میکنیم :

mkdir C:\mosquitto_certs

cd C:\mosquitto_certs

(برای ایجاد کلید خصوصی server.key)

openssl genpkey -algorithm RSA -out server.key -pkeyopt rsa_keygen_bits:2048

(برای ایجاد گواهینامه خود امضاء شده server.crt)

openssl req -new -x509 -key server.key -out server.crt -days 3650

ما باید اطلاعاتی مانند کشور (Country) و نام سازمان (Organization Name) را وارد کنیم.

```
C:\mosquitto_certs>openssl req -new -x509 -key server.key -out server.crt -days 3650
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IRAN
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:IR
State or Province Name (full name) [Some-State]:TEHRAN
Locality Name (eg, city) []:TEHRAN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SIOTALI
Organizational Unit Name (eg, section) []:SIOTCLASS
Common Name (e.g. server FQDN or YOUR name) []:SIOT
Email Address []:aliasadi.ee.iust@gmail.com

C:\mosquitto_certs>
```

تبدیل فایل گواهینامه به فرمت PEM:

```
openssl x509 -in server.crt -out server.pem
```

```
openssl rsa -in server.key -out server.pem
```

```
C:\mosquitto_certs>openssl req -new -x509 -key server.key -out server.crt -days 3650
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IRAN
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:IR
State or Province Name (full name) [Some-State]:TEHRAN
Locality Name (eg, city) []:TEHRAN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SIOTALI
Organizational Unit Name (eg, section) []:SIOTCLASS
Common Name (e.g. server FQDN or YOUR name) []:SIOT
Email Address []:aliasadi.ee.iust@gmail.com

C:\mosquitto_certs>openssl x509 -in server.crt -out server.pem

C:\mosquitto_certs>openssl rsa -in server.key -out server.pem
writing RSA key

C:\mosquitto_certs>
```

Mosquitto: از دستور زیر برای بررسی اینکه Mosquitto روی پورت 8883 در حال گوش دادن است، بررسی وضعیت استفاده میکنیم

```
netstat -an | find "8883"
```

```
C:\Program Files\mosquitto>netstat -an | find "8883"
TCP      0.0.0.0:8883              0.0.0.0:0          LISTENING
TCP      [::]:8883                [::]:0            LISTENING
TCP      [::1]:8883              [::1]:57445        ESTABLISHED
TCP      [::1]:57445              [::1]:8883        ESTABLISHED
```

برای بررسی این که آیا گواهینامه‌ها به درستی ایجاد شده‌اند، می‌توانیم با استفاده از دستور زیر در OpenSSL آن‌ها را بررسی کنیم:

```
openssl x509 -in C:\mosquitto\certs\server.crt -text -noout
```

```

Administrator: Command Prompt
C:\Program Files\mosquitto>openssl x509 -in C:\mosquitto_certs\server.crt -text -noout
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number:
        16:a6:c3:b7:6c:2f:b3:bd:93:08:c9:83:39:2d:2d:51:58:0c:94:5f
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = IR, ST = TEHRAN, L = TEH, O = SIOT, OU = SIOTALI, CN = IOT, emailAddress = aliasadi.ee.iust@gmail.com
    Validity
        Not Before: Dec 25 20:40:50 2024 GMT
        Not After : Dec 23 20:40:50 2034 GMT
    Subject: C = IR, ST = TEHRAN, L = TEH, O = SIOT, OU = SIOTALI, CN = IOT, emailAddress = aliasadi.ee.iust@gmail.com
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
        Modulus:
            00:d8:88:df:bb:cc:1b:f8:cd:d7:fc:48:16:2a:ad:
            8f:b3:e3:a2:30:b5:ad:64:b6:b7:41:b1:92:d1:d5:
            0b:04:a0:08:01:38:4d:95:7b:74:3d:d8:c0:0f:f5:
            fe:72:28:ba:3f:3b:fb:ce:47:90:3f:b7:5d:6a:fd:
            3b:59:8f:4c:97:ab:ee:ec:00:d6:9e:14:8a:28:26:
            05:ac:10:81:3f:1f:08:f2:dc:3c:07:21:b0:85:2e:
            e2:f7:a4:ee:1e:73:36:33:e4:31:b9:03:6b:6d:97:
            7d:9c:65:93:cd:d8:bb:27:b2:10:95:67:89:64:e1:
            0c:c3:02:79:51:6:fd:64:9f:3c:99:1d:67:9e:a2:
            89:52:97:1f:30:58:6b:5e:23:05:0b:cd:e8:72:cc:
            e5:01:4f:cb:26:1f:d3:9f:26:5d:a0:b8:f2:ae:27:
            4f:c8:75:9c:e0:39:f1:93:24:f7:f1:88:ee:e4:ed:
            4b:68:36:e0:14:59:f6:f8:26:3b:34:89:17:7b:86:
            6d:55:52:48:c8:f6:eb:72:42:3d:56:bc:ae:2f:88:
            a0:f5:2d:0b:2f:63:88:37:29:99:87:5c:ba:99:e7:
            d8:48:ee:64:ec:ad:bc:8c:c4:dc:1b:05:5c:0f:0b:
            7a:e8:c6:4c:bf:a1:bd:5f:4d:16:1f:78:a3:09:4d:
            f5:c5
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Subject Key Identifier:
        96:87:CF:8D:B5:05:BC:06:92:A0:B1:C1:27:78:22:DF:65:61:E0:8F
    X509v3 Authority Key Identifier:
        96:87:CF:8D:B5:05:BC:06:92:A0:B1:C1:27:78:22:DF:65:61:E0:8F
    X509v3 Basic Constraints: critical
        CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
60:19:6f:b4:2a:33:47:a3:91:0b:27:05:51:c2:ff:d2:55:3c:
10:11:e4:9b:55:90:37:6f:6b:d6:49:d7:c2:3a:e1:bb:2d:2d:
06:4a:55:b4:8b:81:6c:6a:83:67:ce:99:b7:74:1a:8d:4e:40:
53:b0:f0:23:3f:96:e2:ca:41:8f:a3:4e:6f:99:e8:39:be:f0:
d0:58:a4:79:6f:51:5a:44:3a:18:32:d3:6d:7b:f0:23:44:79:
7a:0e:7d:0b:83:ad:6d:79:b3:f0:7b:8e:68:28:da:19:84:af:
8c:1d:2a:40:1c:56:d5:4c:9d:66:a6:a8:f3:94:41:65:76:1a:
0b:bf:9e:0d:2a:dd:e0:b0:8f:30:0b:1d:6b:30:cb:0d:08:67:
47:24:c5:e8:f3:3f:8d:59:5c:5:f4:fb:88:0d:7f:fe:7f:06:
75:6f:1c:2b:61:32:f1:31:71:18:df:2a:a0:20:7e:e4:3e:67:
50:1b:cc:9b:a4:d9:df:54:c6:11:b1:b2:58:03:0f:cb:b9:b8:
aa:c0:76:61:46:0f:98:56:e7:ce:5a:3a:2f:93:e1:6c:32:22:
8c:58:cc:74:4e:54:c0:84:b9:7a:4a:1e:57:d9:80:a4:48:30:
b6:db:2e:b3:49:cb:5f:07:80:80:20:3d:31:e6:27:3a:ed:62:
3c:09:b1:b7

```

```

X509v3 Authority Key Identifier:
    96:87:CF:8D:B5:05:BC:06:92:A0:B1:C1:27:78:22:DF:65:61:E0:8F
X509v3 Basic Constraints: critical
    CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
60:19:6f:b4:2a:33:47:a3:91:0b:27:05:51:c2:ff:d2:55:3c:
10:11:e4:9b:55:90:37:6f:6b:d6:49:d7:c2:3a:e1:bb:2d:2d:
06:4a:55:b4:8b:81:6c:6a:83:67:ce:99:b7:74:1a:8d:4e:40:
53:b0:f0:23:3f:96:e2:ca:41:8f:a3:4e:6f:99:e8:39:be:f0:
d0:58:a4:79:6f:51:5a:44:3a:18:32:d3:6d:7b:f0:23:44:79:
7a:0e:7d:0b:83:ad:6d:79:b3:f0:7b:8e:68:28:da:19:84:af:
8c:1d:2a:40:1c:56:d5:4c:9d:66:a6:a8:f3:94:41:65:76:1a:
0b:bf:9e:0d:2a:dd:e0:b0:8f:30:0b:1d:6b:30:cb:0d:08:67:
47:24:c5:e8:f3:3f:8d:59:5c:5:f4:fb:88:0d:7f:fe:7f:06:
75:6f:1c:2b:61:32:f1:31:71:18:df:2a:a0:20:7e:e4:3e:67:
50:1b:cc:9b:a4:d9:df:54:c6:11:b1:b2:58:03:0f:cb:b9:b8:
aa:c0:76:61:46:0f:98:56:e7:ce:5a:3a:2f:93:e1:6c:32:22:
8c:58:cc:74:4e:54:c0:84:b9:7a:4a:1e:57:d9:80:a4:48:30:
b6:db:2e:b3:49:cb:5f:07:80:80:20:3d:31:e6:27:3a:ed:62:
3c:09:b1:b7

```

C:\Program Files\mosquitto>

ری استارت:

```
C:\Program Files\mosquitto>net stop mosquitto
The Mosquitto Broker service is not started.

More help is available by typing NET HELPMSG 3521.

C:\Program Files\mosquitto>net start mosquitto
The Mosquitto Broker service is starting.
The Mosquitto Broker service could not be started.

The service did not report an error.

More help is available by typing NET HELPMSG 3534.

C:\Program Files\mosquitto>
```

دستور پایه برای سابسکرایب کردن:

mosquitto_sub -h <hostname> -p <port> -t "<topic>" --cafile <path_to_ca_file> --tls-version tlsv1.2 --insecure

- آدرس هاست یا IP سرور. برای لوکال هاست از localhost یا 127.0.0.1 استفاده کنید.

- شماره پورت. اگر از پورت 8883 (TLS) استفاده می کنید، باید پورت 8883 را وارد کنید.

- نام تاپیک که می خواهید پیامها را از آن دریافت کنید.

--cafile <path_to_ca_file> که برای اعتبارسنجی استفاده می شود. مسیر فایل گواهی CA (certificate authority)

--tls-version tlsv1.2 مشخص کردن نسخه TLS برای اتصال امن.

--insecure این گزینه برای نادیده گرفتن خطاهای گواهی های SSL که ممکن است خود امضا (self-signed) باشند، استفاده می شود.

سابسکرایب کردن با TLS پورت 8883:

mosquitto_sub -h localhost -p 8883 -t "topic_name" --cafile C:\mosquitto_certs\server.crt --tls-version tlsv1.2 --insecure

دستور پایه برای پابلیش:

```
mosquitto_pub -h <hostname> -p <port> -t "<topic>" -m "<message>"
```

- آدرس هاست یا IP سرور MQTT در اینجا می‌توانید از localhost یا 127.0.0.1 استفاده کنید.
- شماره پورت 1883 بدون TLS استفاده می‌کنید، باید پورت 1883 را وارد کنید. اگر از پورت 8883 با TLS استفاده می‌کنید، پورت 8883 را وارد کنید.

- نام تاپیک که می‌خواهید پیام را به آن ارسال کنید.

- پیامی که می‌خواهید ارسال کنید.

پابلیش کردن پیام به تاپیک در پورت 8883 با TLS :

```
mosquitto_pub -h localhost -p 8883 -t "topic_name" -m "Hello, secure MQTT!" --cafile C:\mosquitto_certs\server.crt --tls-version tlsv1.2 --insecure
```

--cafile C:\mosquitto_certs\server.crt مسیر فایل گواهی (certificate) شما که برای اعتبارسنجی استفاده می‌شود.

TLS--tls-version tlsv1.2 مشخص کردن نسخه

-اگر گواهی‌نامه‌ها به صورت خود امضا شده (self-signed) هستند، از این گزینه برای نادیده گرفتن خطاهای گواهی استفاده می‌شود.

همینطور mosquito_certs را باید در Orange Pi هم کپی کنیم تا آنجا هم قابل استفاده باشد در Conf فایل Orange Pi مخصوص خود را دارد

listener 8883

cafile /opt/iot/python/server.pem

certfile /opt/iot/python/server.crt

keyfile /opt/iot/python/server.key

protocol mqtt

تعريف فایل رمز عبور

allow_anonymous false

password_file C:\mosquitto\mosquitto_passwd_file

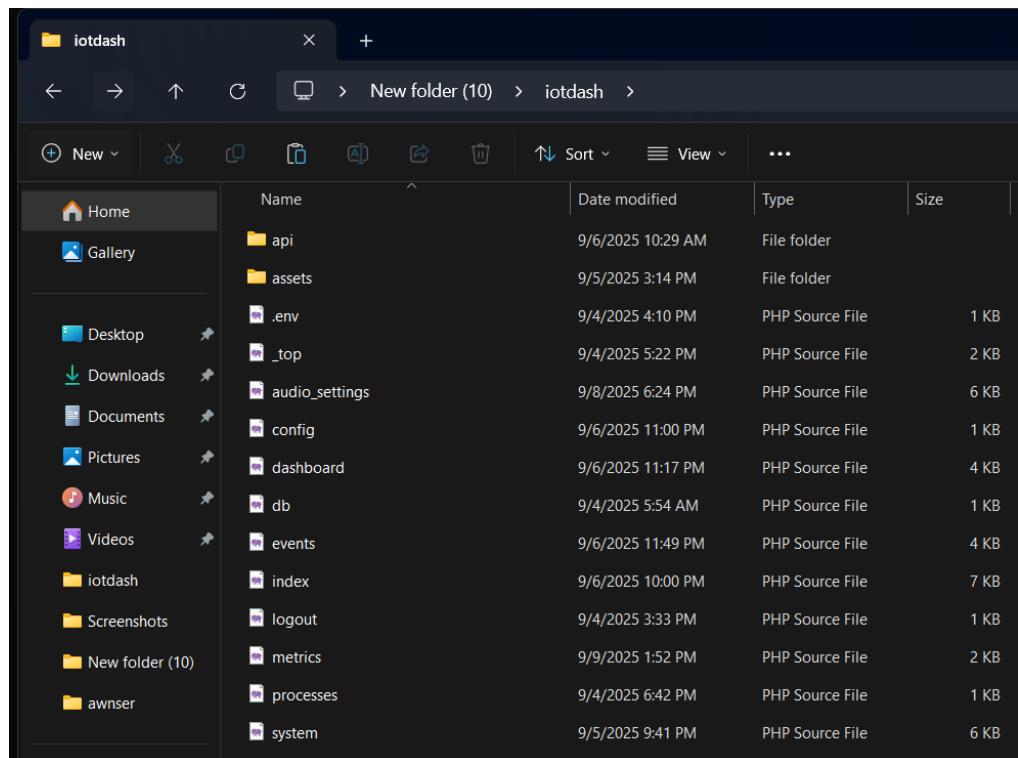
acl_file /opt/iot/python/aclfile

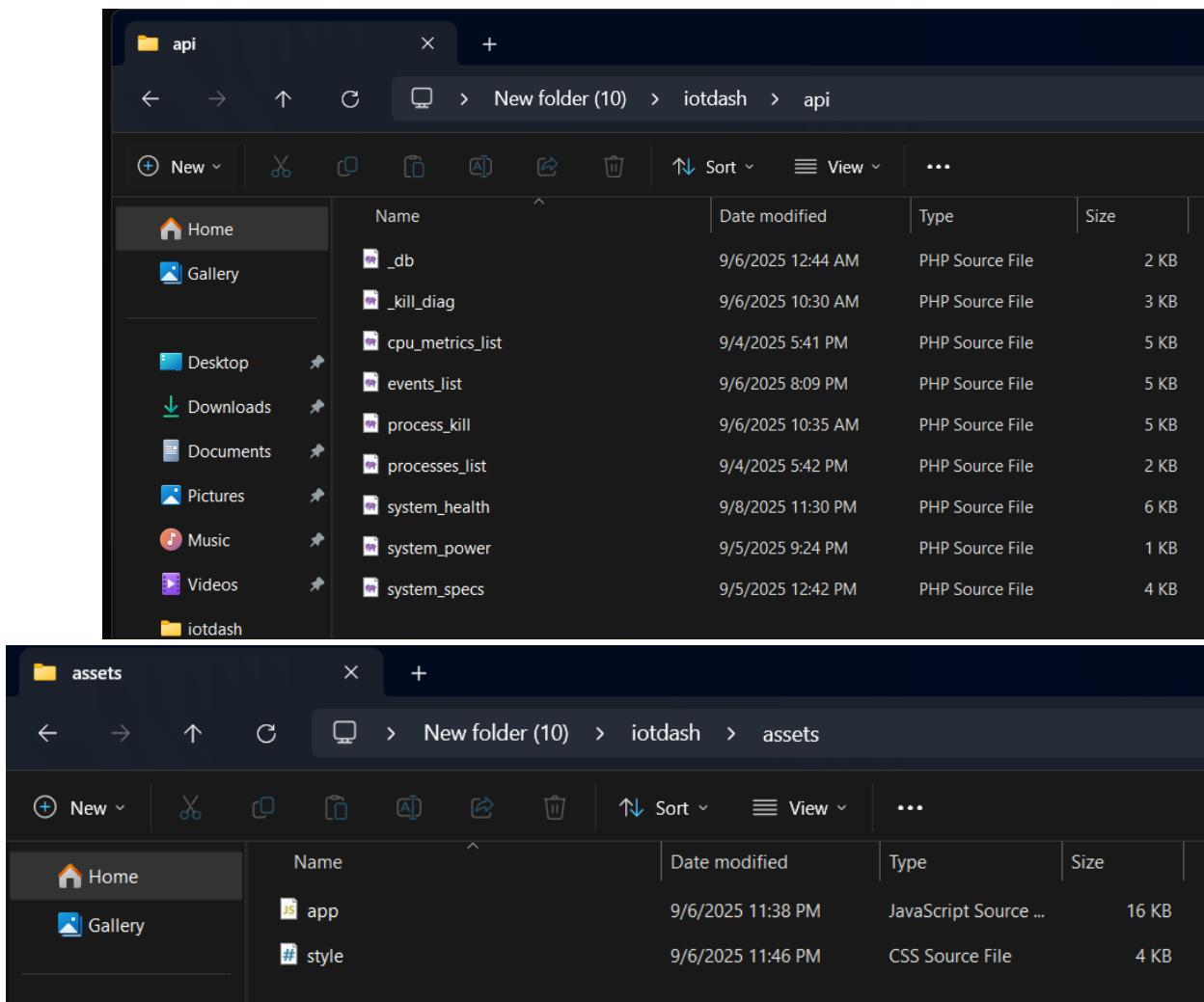
فایل acl هم برای دادن سطح دسترسی تعریف کردیم که در تعریف کدها آن هم توضیح میدهیم

همینطور یوزر مخصوص و پس هم تعریف میکنیم:

```
root@orangepizeroplus:/opt/iot/python# sudo mosquitto_passwd -c /etc/mosquitto/passwd i
otuser
Password: root@orangepizeroplus:/opt/iot/python# ^C
root@orangepizeroplus:/opt/iot/python# ^C
root@orangepizeroplus:/opt/iot/python# sudo mosquitto_passwd -c /etc/mosquitto/passwd Ali
Password:
Reenter password:
Error: Passwords do not match.
root@orangepizeroplus:/opt/iot/python# sudo mosquitto_passwd -c /etc/mosquitto/passwd Ali
Password:
Reenter password:
Error: Passwords do not match.
root@orangepizeroplus:/opt/iot/python# sudo mosquitto_passwd -c /etc/mosquitto/passwd Ali
Password:
Reenter password:
root@orangepizeroplus:/opt/iot/python# sudo systemctl restart mosquitto
Job for mosquitto.service failed because the control process exited with error code.
See "systemctl status mosquitto.service" and "journalctl -xe" for details.
root@orangepizeroplus:/opt/iot/python# sudo systemctl status mosquitto --no-pager -l
```

کد هایی هم که برای تعریف وب تعریف کردیم شامل Rest API ها و کدهای php و یک جاوا اسکریپت و CSS را به اورنج پای منتقل میکنیم:





در اورنج پایی و putty :

```
sudo rm -rf /home/orangepi/iotdash
```

```
mkdir -p /home/orangepi/iotdash
```

```
sudo chown -R orangepi:orangepi /home/orangepi/iotdash
```

در cmd ویندوز :

```
"C:\Program Files\PuTTY\pscp.exe" -r "C:\Users\Ali\Desktop\New folder (10)\iotdash"
orangepi@192.168.100.99:/home/orangepi/
```

در اورنج پایی :

```

sudo mkdir -p /var/www/html/iotdash
sudo cp -r /home/orangepi/iotdash/* /var/www/html/iotdash/
sudo chown -R www-data:www-data /var/www/html/iotdash

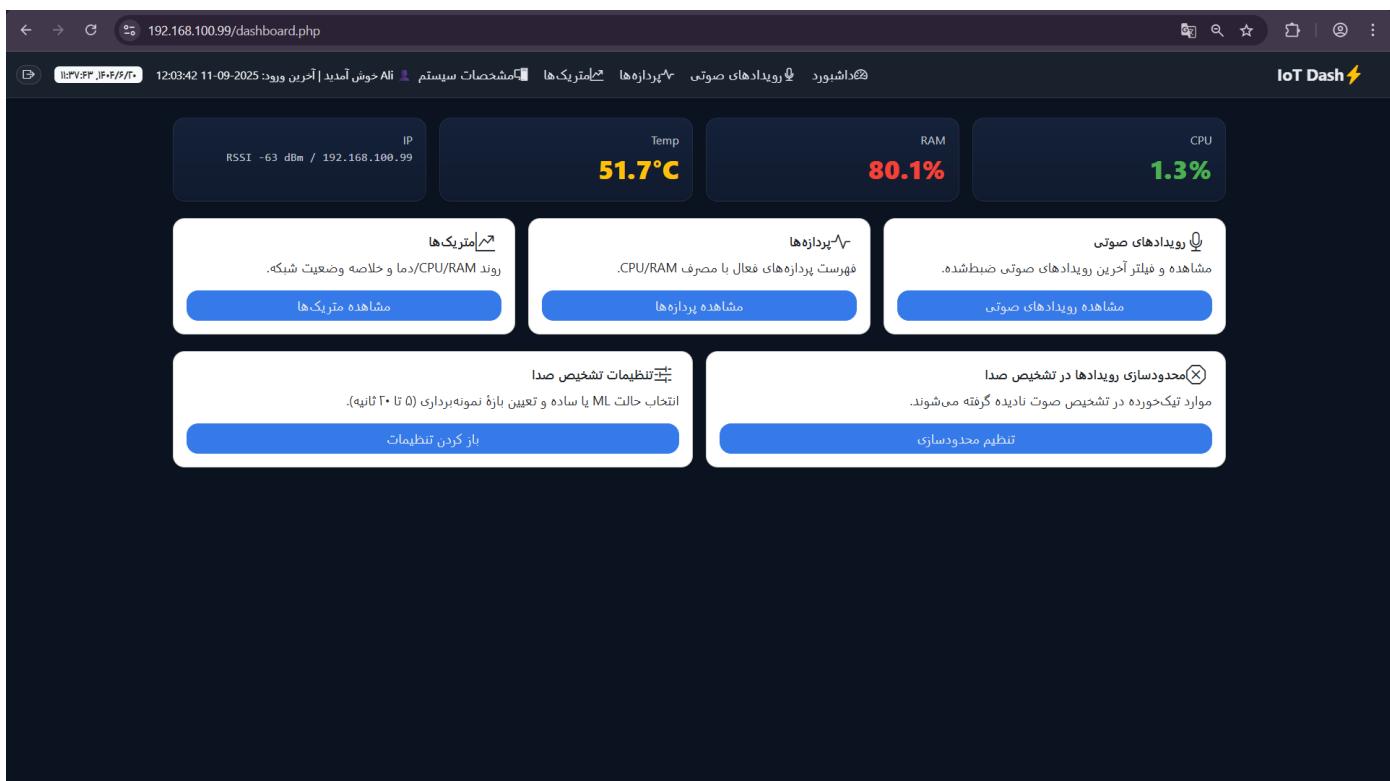
```

مروگر لپتاپ:

<http://192.168.100.99/iotdash/index.php>

Ali یوزر

پسورد EmbeddedProject



برای این سازی سایت چندین راه هست
ابتدا با برنامه mkcert.exe و دستور زیر در cmd ویندوز

mkcert.exe 192.168.100.99 iotdash.local

```

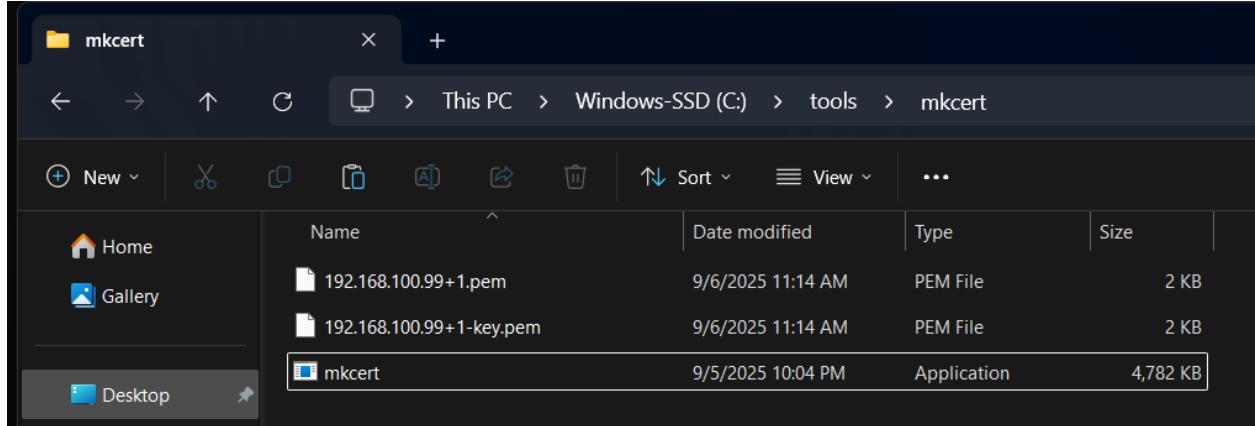
PS C:\tools\mkcert> C:\tools\mkcert\mkcert.exe 192.168.100.99 iotdash.local

Created a new certificate valid for the following names 🌐
- "192.168.100.99"
- "iotdash.local"

The certificate is at "./192.168.100.99+1.pem" and the key at "./192.168.100.99+1-key.pem" ✅
It will expire on 6 December 2027 📆

PS C:\tools\mkcert>

```



مکررت (mkcert) یک گواهی SSL/TLS خودامضا (self-signed) تولید کرده که برای دو اسم معتبره:

1. آدرس آیپی 192.168.100.99

2. نام دامنه‌ی لوکال iotdash.local

فایل گواهی (certificate) ساخته شده:

./192.168.100.99+1.pem

کلید خصوصی (private key) ساخته شده:

./192.168.100.99+1-key.pem

این گواهی تا تاریخ 6 دسامبر 2027 معتبر خواهد بود.

این گواهی رو می‌تونی در وب‌서ور (Apache/Nginx) روی Orange Pi قرار بدیم تا به جای HTTP معمولی، با HTTPS امن کار کنه.

مثلاً مرورگر وقتی با /iotdash.local/ یا https://192.168.100.99/ وصل بشه، همین گواهی رو می‌بینه.

برای [SVC-SYSTEMD] کد های بالا در زمان ریبوت سیستم یا هر بار بالا آمدن سیستم.

```
sudo usermod -aG audio orangepi
```

: (resource_monitor_mqtt.py) مانیتور منابع

```
sudo tee /etc/systemd/system/iot-resource-monitor.service >/dev/null <<'UNIT'
```

```
[Unit]
```

```
Description=IoT Resource Monitor (MQTT)
```

```
After=network-online.target
```

```
Wants=network-online.target
```

```
[Service]
```

```
Type=simple
```

```
WorkingDirectory=/opt/iot/python
```

```
ExecStart=/usr/bin/python3 /opt/iot/python/resource_monitor_mqtt.py
```

```
Restart=always
```

```
RestartSec=3
```

```
Environment=PYTHONUNBUFFERED=1
```

```
# محدودیت های سبک اختیاری
```

```
# Nice=5
```

```
# CPUQuota=80%
```

```
# سخت سازی پیشنهادی
```

```
NoNewPrivileges=true
```

```
ProtectSystem=full
```

```
ReadWritePaths=/opt/iot
```

```
ProtectHome=true
```

```
PrivateTmp=true
```

[Install]

WantedBy=multi-user.target

UNIT

: (detect_audio_yamnet_db_ML.py) تشخیص صدا

sudo tee /etc/systemd/system/iot-detect-audio.service >/dev/null <<'UNIT'

[Unit]

Description=Audio Event Detection (YAMNet)

After=network-online.target sound.target mysql.service mariadb.service

Wants=network-online.target sound.target

[Service]

Type=simple

WorkingDirectory=/opt/iot/python

ExecStart=/usr/bin/python3 /opt/iot/python/detect_audio_yamnet_db_ML.py

Restart=always

RestartSec=5

Environment=PYTHONUNBUFFERED=1

After=mosquitto.service

Wants=mosquitto.service

ساختسازی پیشنهادی

NoNewPrivileges=true

ProtectSystem=full

ReadWritePaths=/opt/iot

ProtectHome=true

```
PrivateTmp=true  
# StartLimitIntervalSec=0
```

[Install]

```
WantedBy=multi-user.target
```

UNIT

همینطور از [ML-ADV] Advanced ML model beyond threshold/frame differencing استفاده کردیم یعنی کد تمرین ۳ را ارتقا دادیم با این ویژگی ها :

به جای تصمیم لحظه‌ای، احتمال کلاس‌ها را نرم‌سازی و میانگین‌گیری می‌کنیم.

با هیسترزیس + حداقل مدت تنها رخدادهایی تأیید می‌شوند که مدتی پایدار باشند.

با **Entropy gating** خروجی‌های نامطمئن مدل حذف می‌شوند.

با **SNR gate** سکوت/پس‌زمینه حذف می‌شود.

برای کلاس‌های پس‌زمینه (.../Rustle/Hum) داریم و می‌توانی allowed_classes را برای سناریوی خودمان محدود کنیم.

فقط **threshold** ساده روی انرژی یا **frame difference** نیست که معمولاً توی baseline-ها برای صدا/تصویر استفاده می‌شود.

از یک مدل یادگیری ماشین عمیق (YAMNet / TensorFlow) برای استخراج ویژگی و دسته‌بندی استفاده می‌کنیم.

روش‌های **post-processing** پیش‌رفته داره:

- نرم‌سازی زمانی (EMA)
- هیسترزیس on/off
- حداقل مدت رخداد (debounce)
- فیلتر عدم قطعیت با entropy
- فیلتر SNR برای نویز/سکوت

خروجی top-3 کلاس‌ها و ثبت در دیتابیس → باعث می‌شود تحلیل عمیق‌تری نسبت به "آیا تغییری در فریم رخ داد یا نه" داشته باشیم.



[1] Audio Capture (sounddevice)

- ضبط: `sd.rec() ← DURATION_S, RECORD_SR`
- خروجی: `mono (float32)`

[2] SNR Gate (فیلتر سکوت/نویز)

- $\text{snr_db}(\text{mono}) \geq \text{SNR_MIN_DB}$ ؟

و ادامه‌ی حلقه \rightarrow نه —

بله \rightarrow ادامه

[3] Resample 16k برای YAMNet

- اگر `RECORD_SR != TARGET_SR`:

```
mono = resample_poly(... → 16000Hz)
```



[4] Inference لـ YAMNet (TF)

- waveform = tf.convert_to_tensor(mono)
- scores, embeddings, spectrogram = yamnet(waveform)
- frame_probs = softmax(scores) # شکل: [frames, n_classes]



[5] Temporal Averaging + EMA

- probs_mean = mean(frame_probs, axis=0) # میانگین روی فریم‌ها
- probs_smooth = EMA.update(probs_mean) # TemporalSmoothen(a=0.65)



[6] Uncertainty Gate (Entropy)

- H = softmax_entropy(probs_smooth)
- H ≤ ENTROPY_MAX ؟

و ادامه‌ی حلقه skip → نه —

بله → ادامه



[7] EventDetector (هیسترزیس + حداقل مدت)

- top class = $\text{argmax}(\text{probs_smooth})$, conf = $\text{max}(\text{probs_smooth})$

- کلاس‌های ممنوع/پس‌زمینه حذف می‌شوند (ban_background)

- منطق هیسترزیس:

برای ورود $\text{on_th} = 0.40$

برای خروج $\text{off_th} = 0.25$

- $\text{min_dur_s} = 1.0$ باید حداقل ~ 2 فریم پایدار بماند \rightarrow

- آماده $\rightarrow \text{emitted=True}$ و $(\text{label}, \text{conf}, \text{top3})$ شرایط برقرار شد



`emitted ? _____`



No ▼

| Yes



چاپ no-event [8] Emit Event

- $\text{timestamp} = \text{now}()$

- چاپ اطلاعات رخداد

- extra = {top3, entropy}



▼ [9] DB Insert (PyMySQL)

- `safe_insert(label, conf, ts, extra)`

ریکانکت خودکار در خطا -

|
|
▼ بازگشت به ابتدای حلقه

حذف سکوت/پس زمینه‌ی بسیار کم قدرت—قبل از هزینه‌ی محاسباتی مدل.

مدل یادگیری عمیق از گوگل (16000Hz)، خروجی امتیاز کلاس‌ها برای فریم‌ها.

کاهش نوسانات لحظه‌ای؛ سیگنال احتمال را هموار می‌کند.

وقتی توزیع احتمال خیلی «پهن» است (عدم قطعیت بالا)، رخداد رد می‌شود.

EventDetector (Hysteresis + Min-Duration + Banlist)

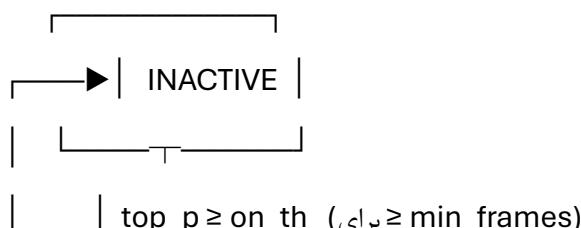
هیسترزیس: دو آستانه‌ی ورود/خروج برای جلوگیری از پرش.

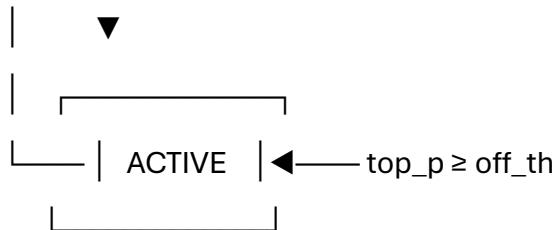
حداقل مدت: باید چند فریم متوالی بالای آستانه‌ی ورود بماند تا رخداد تأیید شود.

کلاس‌های پس زمینه مثل «Hum/Rustle/Noise/...» حذف می‌شوند.

درج رخداد با `raw_json` و ریکانکت در خطا.

منطق هیسترزیس State Machine ساده





$\text{top_p} < \text{off_th} \rightarrow \text{back to INACTIVE}$

on_th/off_th, min_dur_s طول هر بافر ضبط DURATION_S

ENTROPY_MAX (، EMA) شدت سختگیری عدم قطعیت SMOOTH_ALPHA

حساسیت سکوت/نویز SNR_MIN_DB

برای سناریوهای خاص allowed_classes یا ban_background

همینطور از داکر استفاده کردیم

sudo apt update

sudo apt install -y \

ca-certificates curl gnupg lsb-release

```
COM3 - PuTTY
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
37 added, 20 removed; done.
Setting up gnupg-110n (2.2.19-3ubuntu2.5) ...
Setting up gpgconf (2.2.19-3ubuntu2.5) ...
Setting up libcurl4:arm64 (7.68.0-1ubuntu2.25) ...
Setting up curl (7.68.0-1ubuntu2.25) ...
Setting up gpg (2.2.19-3ubuntu2.5) ...
Setting up gnupg-utils (2.2.19-3ubuntu2.5) ...
Setting up gpg-agent (2.2.19-3ubuntu2.5) ...
Setting up gpgsm (2.2.19-3ubuntu2.5) ...
Setting up dirmngr (2.2.19-3ubuntu2.5) ...
Setting up gpg-wks-server (2.2.19-3ubuntu2.5) ...
Setting up gpg-wks-client (2.2.19-3ubuntu2.5) ...
Setting up gnupg (2.2.19-3ubuntu2.5) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for ca-certificates (20240203~20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.

Progress: [ 98% ] #####
```

```

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg | sudo
gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

```

COM3 - PuTTY

plugin arm64 2.35.1-1~ubuntu.20.04~focal [12.0 MB]
Fetched 65.7 MB in 25s (2,642 kB/s)
(Reading database ... 41812 files and directories currently installed.)
Preparing to unpack .../libseccomp2_2.5.1-1ubuntu1~20.04.2_arm64.deb ...
Unpacking libseccomp2:arm64 (2.5.1-1ubuntu1~20.04.2) over (2.4.3-1ubuntu3.20.04.3) ...
Setting up libseccomp2:arm64 (2.5.1-1ubuntu1~20.04.2) ...
Selecting previously unselected package containerd.io.
(Reading database ... 41812 files and directories currently installed.)
Preparing to unpack .../containerd.io_1.7.27-1_arm64.deb ...
Unpacking containerd.io (1.7.27-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../docker-ce-cli_5%3a28.1.1-1~ubuntu.20.04~focal_arm64.deb ...
Unpacking docker-ce-cli (5:28.1.1-1~ubuntu.20.04~focal) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../docker-ce_5%3a28.1.1-1~ubuntu.20.04~focal_arm64.deb ...
Unpacking docker-ce (5:28.1.1-1~ubuntu.20.04~focal) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../docker-compose-plugin_2.35.1-1~ubuntu.20.04~focal_arm64.deb ...
Unpacking docker-compose-plugin (2.35.1-1~ubuntu.20.04~focal) ...

Progress: [ 52%] [#####
root@orangepizeroplus:~# cd /opt/iot/python
root@orangepizeroplus:/opt/iot/python#

```

COM3 - PuTTY

GNU nano 4.8 requirements.txt Modified

```
numpy==1.26.*  
tensorflow==2.11.0  
sounddevice  
scipy  
pymysql  
paho-mqtt
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text^T To Spell ^ Go To Line

COM3 - PuTTY

GNU nano 4.8 Dockerfile Modified

```
FROM python:3.10-slim

# کتابخانه‌های سیستمی لازم برای sounddevice/ALSA
RUN apt-get update && apt-get install -y --no-install-recommends \
    libportaudio2 libasound2 libsndfile1 \
&& rm -rf /var/lib/apt/lists/*

WORKDIR /app

# اول نیازمندیها تا کش build بهتر کار کند
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

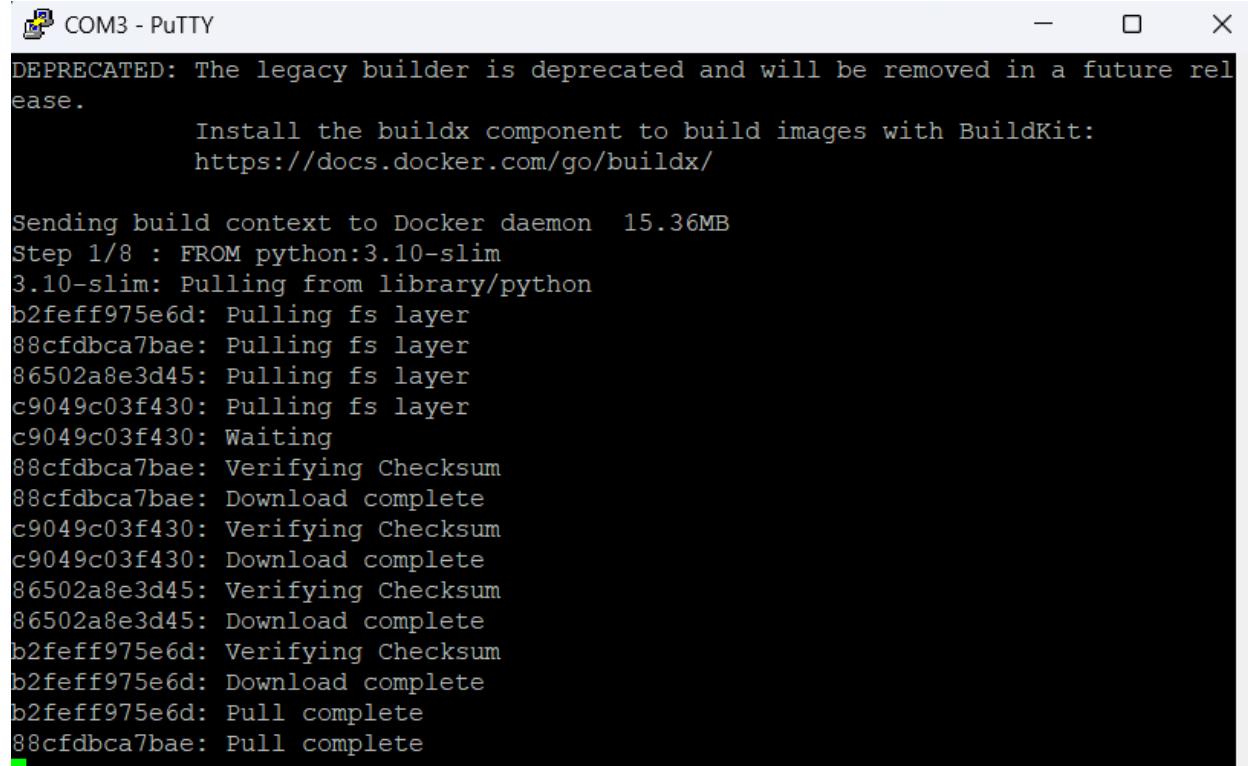
# سپس کل پروژه (مدلها و فایلهای py.
COPY .

# مقادیر پیشفرض محیط
ENV USB_INDEX=2 \
    DB_HOST=localhost \
    DB_USER=iotapp \
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text^T To Spell ^ Go To Line

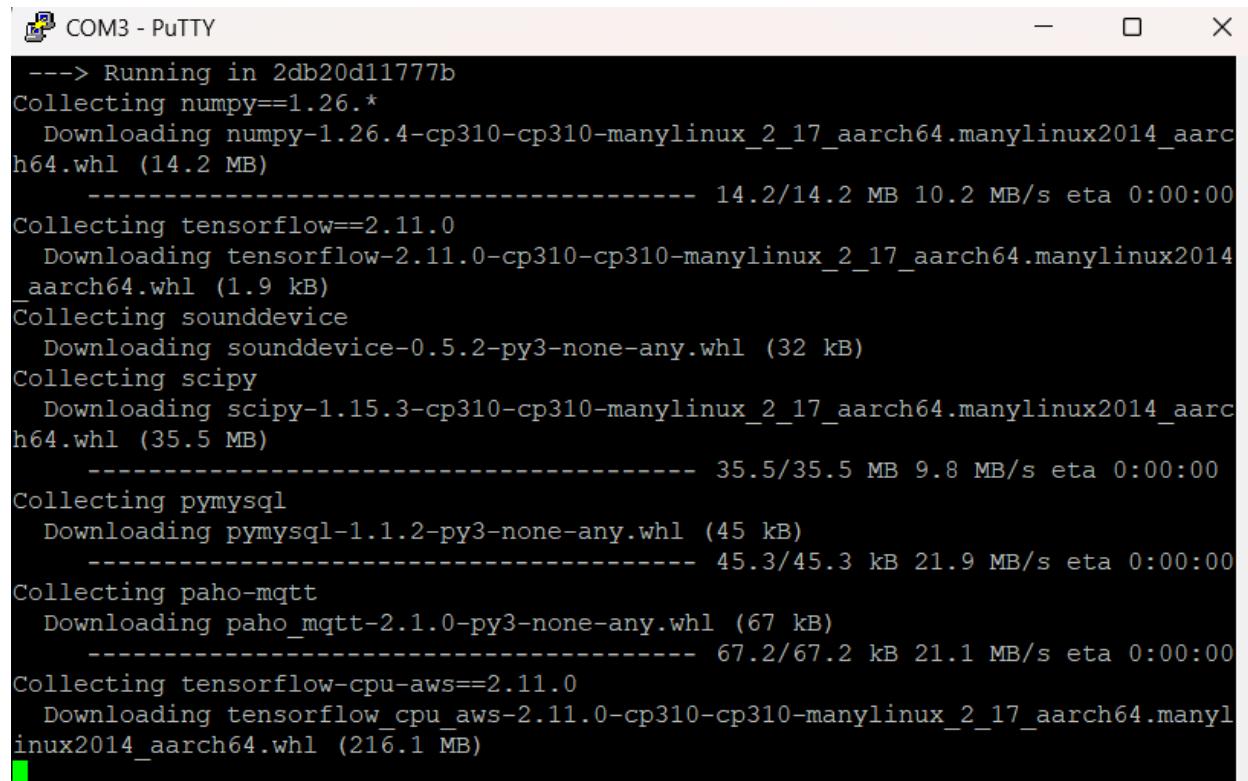
```
cd /opt/iot/python
sudo docker build -t iot-audio .
```

```
root@orangepizeroplus:/opt/iot/python# sudo docker build -t iot-audio .
```



```
COM3 - PuTTY
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 15.36MB
Step 1/8 : FROM python:3.10-slim
3.10-slim: Pulling from library/python
b2feff975e6d: Pulling fs layer
88cfdbca7bae: Pulling fs layer
86502a8e3d45: Pulling fs layer
c9049c03f430: Pulling fs layer
c9049c03f430: Waiting
88cfdbca7bae: Verifying Checksum
88cfdbca7bae: Download complete
c9049c03f430: Verifying Checksum
c9049c03f430: Download complete
86502a8e3d45: Verifying Checksum
86502a8e3d45: Download complete
b2feff975e6d: Verifying Checksum
b2feff975e6d: Download complete
b2feff975e6d: Pull complete
88cfdbca7bae: Pull complete
```



```
COM3 - PuTTY
---> Running in 2db20d11777b
Collecting numpy==1.26.*
  Downloading numpy-1.26.4-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (14.2 MB)
----- 14.2/14.2 MB 10.2 MB/s eta 0:00:00
Collecting tensorflow==2.11.0
  Downloading tensorflow-2.11.0-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (1.9 kB)
Collecting sounddevice
  Downloading sounddevice-0.5.2-py3-none-any.whl (32 kB)
Collecting scipy
  Downloading scipy-1.15.3-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (35.5 MB)
----- 35.5/35.5 MB 9.8 MB/s eta 0:00:00
Collecting pymysql
  Downloading pymysql-1.1.2-py3-none-any.whl (45 kB)
----- 45.3/45.3 kB 21.9 MB/s eta 0:00:00
Collecting paho-mqtt
  Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
----- 67.2/67.2 kB 21.1 MB/s eta 0:00:00
Collecting tensorflow-cpu-aws==2.11.0
  Downloading tensorflow_cpu_aws-2.11.0-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (216.1 MB)
```

```
COM3 - PuTTY

  Downloading urllib3-2.5.0-py3-none-any.whl (129 kB)
  ----- 129.8/129.8 kB 5.5 MB/s eta 0:00:00
Collecting charset_normalizer<4,>=2
  Downloading charset_normalizer-3.4.3-cp310-cp310-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl (147 kB)
  ----- 147.2/147.2 kB 5.1 MB/s eta 0:00:00
Collecting MarkupSafe>=2.1.1
  Downloading MarkupSafe-3.0.2-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (21 kB)
Collecting pyasn1<0.7.0,>=0.6.1
  Downloading pyasn1-0.6.1-py3-none-any.whl (83 kB)
  ----- 83.1/83.1 kB 8.1 MB/s eta 0:00:00
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.3.1-py3-none-any.whl (160 kB)
  ----- 160.1/160.1 kB 4.0 MB/s eta 0:00:00
Installing collected packages: tensorboard-plugin-wit, libclang, flatbuffers, wrapt, urllib3, typing-extensions, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard-data-server, six, pymysql, pycparser, pyasn1, protobuf, paho-mqtt, packaging, opt-einsum, oauthlib, numpy, MarkupSafe, markdown, keras, idna, grpcio, gast, charset_normalizer, certifi, cachetools, absl-py, werkzeug, scipy, rsa, requests, pyasn1-modules, h5py, google-pasta, CFFI, astunparse, sounddevice, requests-oauthlib, google-auth, google-auth-oauthlib, tensorboard, tensorflow-cpu-aws, tensorflow
```

```
--> d5ee0e372660
Step 6/8 : COPY . .
--> 3556e26ac97b
Step 7/8 : ENV USB_INDEX=2           DB_HOST=localhost           DB_USER=iotapp           DB_PASS=
ChangeMe_Strong!           DB_NAME=iotdash
--> Running in ad93678bde10
--> Removed intermediate container ad93678bde10
--> b99a6b956d1e
Step 8/8 : CMD ["python3", "detect_audio_yamnet_db.py"]
--> Running in 07801fe17ce5
--> Removed intermediate container 07801fe17ce5
--> 8b81f6f502cb
Successfully built 8b81f6f502cb
Successfully tagged iot-audio:latest
```

```
sudo docker save -o /opt/iot/iot-audio.tar iot-audio
```

حالا با دستور های زیر میتوانیم داکر را اجرا کنیم :

```
sudo docker run --rm -it \  
  --device /dev/snd \  
  --network host \  
  -e USB_INDEX=2 \  
  ./alsa-mixer
```

```
-e DB_HOST=127.0.0.1 -e DB_USER=Ali -e DB_PASS=EmbeddedProject' -e  
DB_NAME=iotdash \  
iot-audio
```

کانتینر مثل یه «جعبه ایزوله» هست که داخلش برنامه + کتابخونه‌ها + وابستگی‌هاش رو بسته‌بندی می‌کنه.

این جعبه روی هر سیستمی که Docker نصب باشه همون طور اجرا میشه، بدون اینکه مهم باشه سیستم اصلی چه نسخه‌ای از لینوکس یا چه پکیج‌هایی داره.

ما می‌ایم یکی از این بخش‌ها (مثلًا اسکریپت تشخیص صدارو داخل یک کانتینر Docker اجرا می‌کنیم).

اون کانتینر شامل Python + TensorFlow + کتابخونه‌های صوتی + خود اسکریپت هست.

دیگه مهم نیست روی برد چه نسخه‌ای از TensorFlow یا Python نصب باشه، چون همه‌چی داخل کانتینر بسته‌بندی شده.

مزایای استفاده از Docker اینجا

۱. ایزوله‌سازی: اگه اسکریپت قفل کرد یا کرش کرد، روی سیستم اصلی تاثیر کمتری داره.

قابلیت حمل: همون کانتینر رو می‌تونیم روی لپ‌تاپ یا هر سرور دیگه ران کنیم → بدون نصب دوباره همه‌ی پکیج‌ها.

مدیریت راحت: با یک دستور docker start/stop اسکریپت بالا و پایین می‌اد.

داکر تو پروژه‌ی مثل یک ظرف بسته‌بندی برای اسکریپت تشخیص صدا عمل می‌کنه.

مزیتش اینه که همه‌ی وابستگی‌ها (Python, TF, numpy, ...) داخل خودش هستن.

ما سرویس تشخیص صدا را به صورت یک کانتینر Docker پیاده‌سازی کردیم تا قابلیت حمل و ایزوله‌سازی فراهم شود

: UFW برای

```
sudo apt update
```

```
sudo apt install ufw -y
```

```
sudo ufw enable
```

```
(Reading database ... 42045 files and directories currently installed.)  
Preparing to unpack .../ufw_0.36-6ubuntu1.1_all.deb ...  
Unpacking ufw (0.36-6ubuntu1.1) ...  
Setting up ufw (0.36-6ubuntu1.1) ...  
  
Creating config file /etc/ufw/before.rules with new version  
  
Creating config file /etc/ufw/before6.rules with new version  
  
Creating config file /etc/ufw/after.rules with new version  
  
Creating config file /etc/ufw/after6.rules with new version  
Created symlink /etc/systemd/system/multi-user.target.wants/ufw.service → /lib/systemd/system/ufw.service.  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for rsyslog (8.2001.0-1ubuntu1.1) ...  
Processing triggers for systemd (245.4-4ubuntu3.4) ...  
root@orangepizeroplus:~# which ufw  
/usr/sbin/ufw  
root@orangepizeroplus:~# sudo ufw status  
Status: inactive  
root@orangepizeroplus:~# sudo ufw enable  
Firewall is active and enabled on system startup  
root@orangepizeroplus:~#
```

جازه به SSH

```
sudo ufw allow 2222/tcp
```

جازه به وب سرور

```
sudo ufw allow 80/tcp
```

```
sudo ufw allow 443/tcp
```

جازه به MQTT مثلًا پورت 1883 یا 8883 برای TLS

```
sudo ufw allow 8883/tcp
```

COM3 - PuTTY

```
Processing triggers for rsyslog (8.2001.0-1ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
root@orangepireoplus:~# which ufw
/usr/sbin/ufw
root@orangepireoplus:~# sudo ufw status
Status: inactive
root@orangepireoplus:~# sudo ufw enable
Firewall is active and enabled on system startup
root@orangepireoplus:~# sudo ufw allow ssh
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 80/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 443/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 8883/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~#
```

COM3 - PuTTY

```
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 443/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw allow 8883/tcp
Rule added
Rule added (v6)
root@orangepireoplus:~# sudo ufw status
Status: active

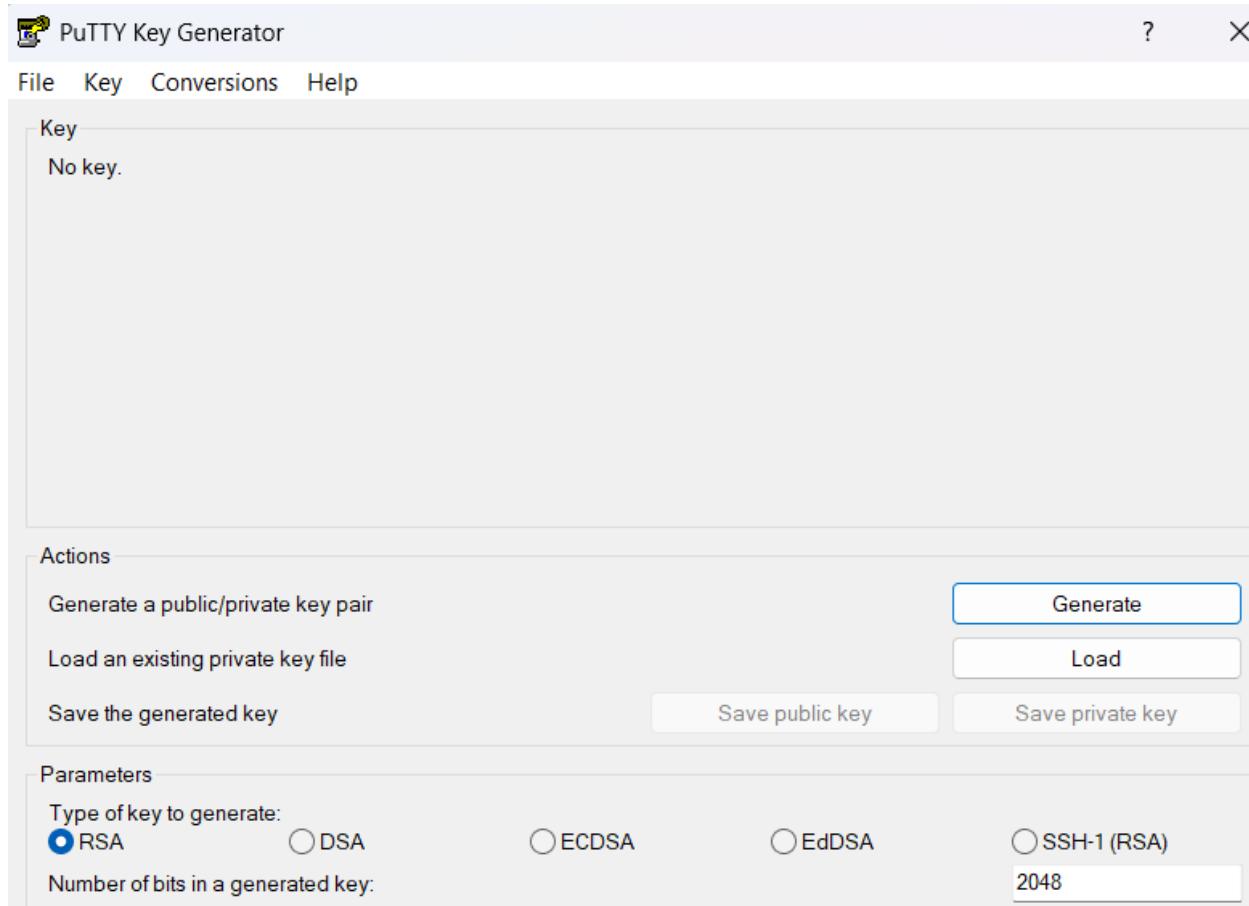
To                         Action      From
--                         -----      ---
22/tcp                      ALLOW       Anywhere
2222/tcp                     ALLOW       Anywhere
80/tcp                       ALLOW       Anywhere
443/tcp                      ALLOW       Anywhere
8883/tcp                     ALLOW       Anywhere
22/tcp (v6)                  ALLOW       Anywhere (v6)
2222/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                   ALLOW       Anywhere (v6)
443/tcp (v6)                 ALLOW       Anywhere (v6)
8883/tcp (v6)                ALLOW       Anywhere (v6)

root@orangepireoplus:~#
```

SSH with keys:

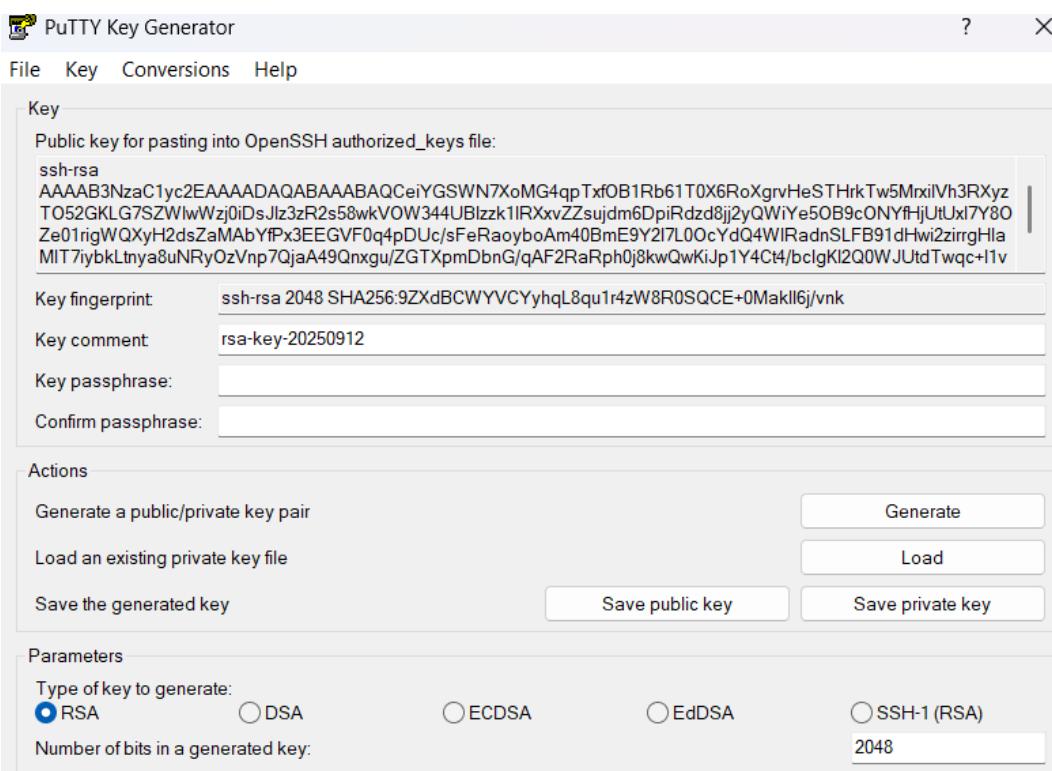
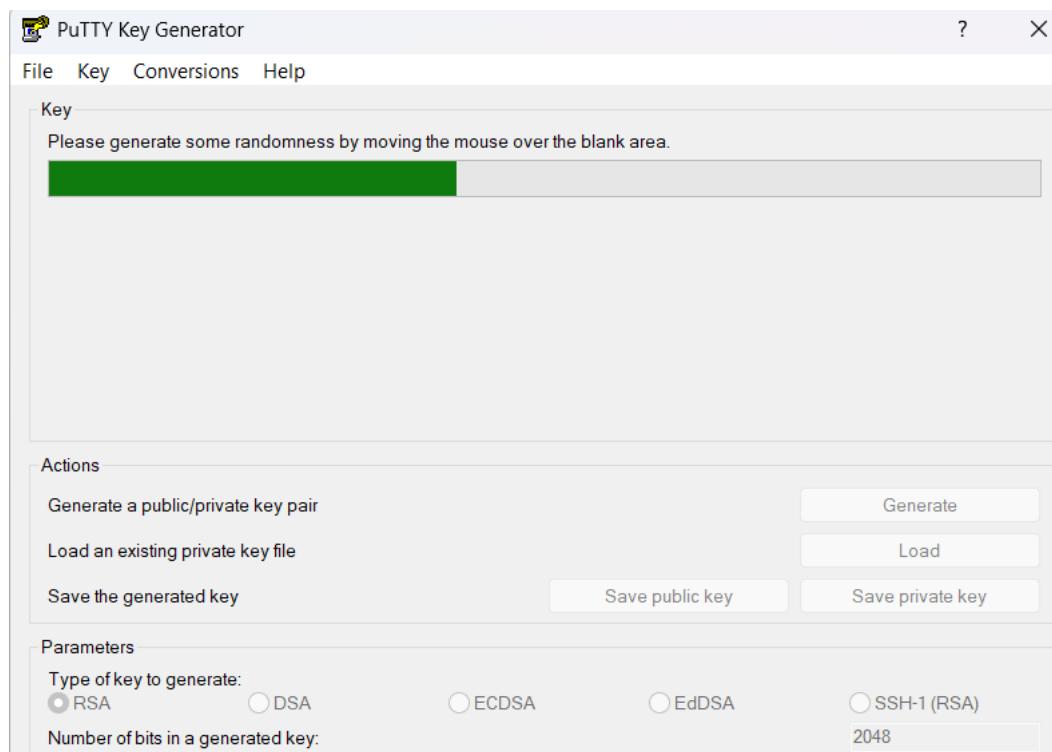
ساختن کلید

1. توی ویندوز **PuTTYgen** رو باز میکنیم



2. گزینه **RSA** یا **ED25519** رو انتخاب میکنیم ED25519 جدیدتر و امن تره

3. روی **Generate** میزنیم و موس رو تكون میدیم تا تصادفی سازی کامل بشه.



: حالاً .4

بخش **Public key for pasting** میکنیم و بعداً می‌ذاریم توی برد (Orange Pi) اینو کپی می‌کنیم

روی **Save private key** میزینم و فایل خصوصی مثلًا `id_rsa.ppk` ذخیره میکنیم (این فقط روی لپ تاپ خودمون میمونه).

اضافه کردن کلید عمومی به برد

1. با پسورد معمولیمون به بردمون وصل بشیم:

```
ssh user@192.168.100.99 .2
```

3. توی برد مطمئن میشیم پوشه SSH وجود داره:

```
mkdir -p ~/.ssh .4
```

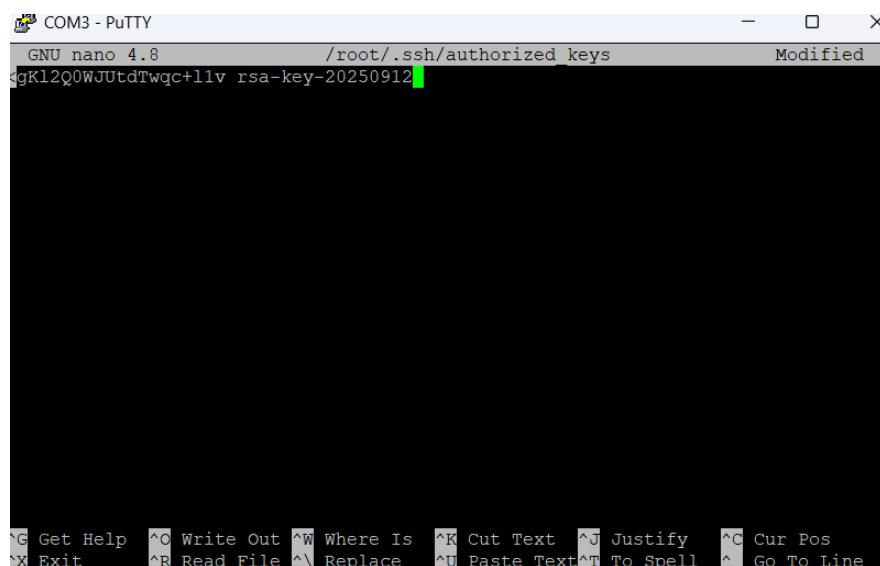
```
chmod 700 ~/.ssh .5
```

```
root@orangepizeroplus:~# mkdir -p ~/.ssh  
root@orangepizeroplus:~# chmod 700 ~/.ssh  
root@orangepizeroplus:~# █
```

6. فایل کلید عمومی رو باز میکنیم:

```
nano ~/.ssh/authorized_keys .7
```

8. کلید عمومی که از PuTTYgen کپی کردیم (ون متن طولانی که با `ssh-ed25519` یا `rsa` شروع میشه) رو اینجا پیست میکنیم و ذخیره میکنیم.



```
COM3 - PuTTY  
GNU nano 4.8 /root/.ssh/authorized_keys Modified  
gKl2q0WJUtdTwqc+l1v rsa-key-20250912 █  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^L Go To Line
```

9. دسترسی‌ها رو درست میکنیم:

```
chmod 600 ~/.ssh/authorized_keys .10
```

```
root@orangepizeroplus:~# chmod 600 ~/.ssh/authorized_keys  
root@orangepizeroplus:~#
```

تنظیم PuTTY برای استفاده از کلید

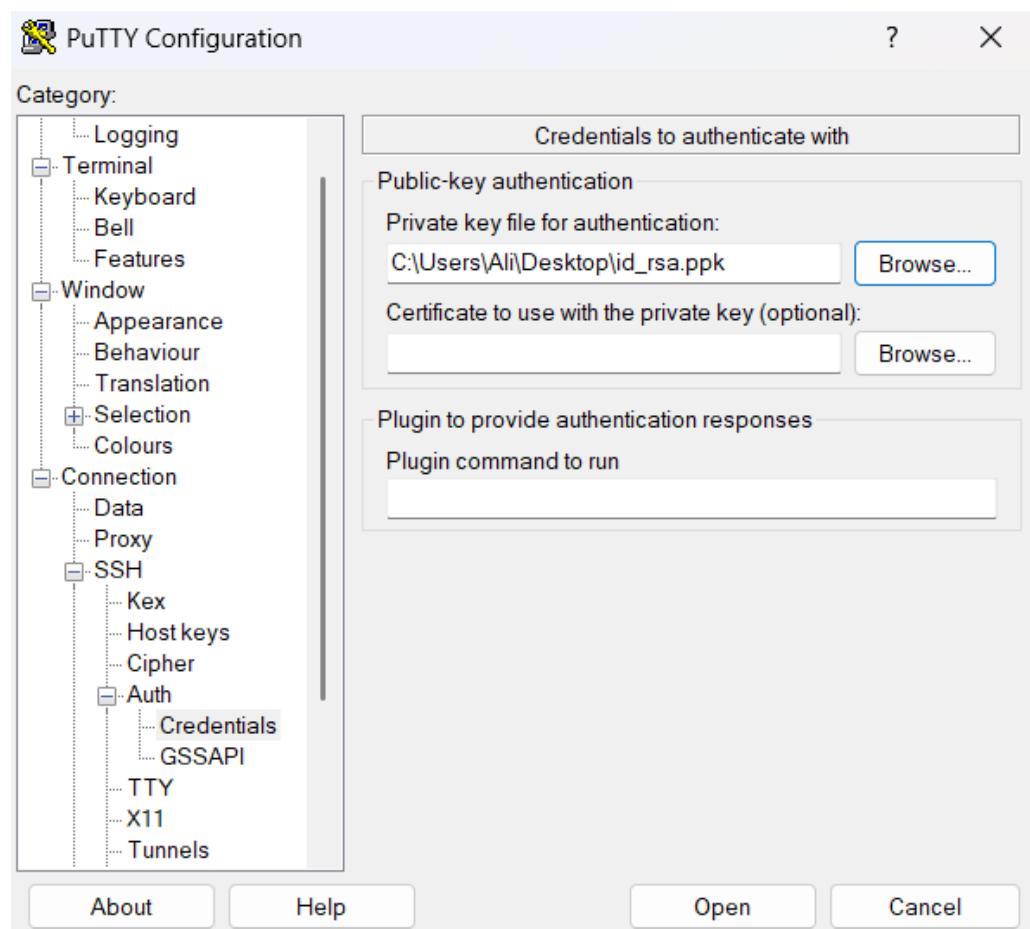
.1. رو باز میکنیم.

.2. توی بخش Host Name آیپی بدمون رو بزنیم 192.168.100.99

.3. در سمت چپ بریم به:

Connection → SSH → Auth → Credentials .4

ونجا روی Browse و فایل خصوصی id_rsa.ppk از رو انتخاب کنیم.



.5. برگردیم به صفحه اصلی PuTTY و روی Open کلیک کنیم.

.6. حالا وقتی وصل بشیم، پسورد نمی‌خواهد و مستقیم با کلیدمن وارد میشه.

```
root@orangepizeroplus: ~
login as: root
Authenticating with public key "rsa-key-20250912"
[REDACTED]
```

برای ali و غیر فعال سازی root

```
ali@orangepizeroplus: ~
login as: ali
Authenticating with public key "rsa-key-20250912"
[REDACTED]
```

وقتی همه چیز درست شد، توی برد می تونیم توی فایل /etc/ssh/sshd_config پسورد رو کلاً غیرفعال کنیم
(PasswordAuthentication no)

: minimal DB access برای
ما در کدمان :

-- ساخت کاربر فقط روی localhost

CREATE USER 'iotapp'@'localhost' IDENTIFIED BY ChangeMe_Strong!';

-- برای خواندن و نوشتن events دادن دسترسی فقط به جدول

GRANT SELECT, INSERT ON iotdash.events TO 'iotapp'@'localhost';

-- اعمال تغییرات

FLUSH PRIVILEGES;

iotdash نام دیتابیس

یعنی:

کاربر دیتابیس جدا باید بسازیم فقط برای اپلیکیشن نه اینکه با root وصل بشه مثلاً کاربر IoTapp فقط اجازه داره روی جدول events عملیات SELECT و INSERT انجام بده. اما اجازه‌ی DROP, DELETE, GRANT, CREATE USER یا تغییر ساختار کل دیتابیس رو نداره.

دسترسی فقط از لوکال‌هوست

یعنی کاربر دیتابیس فقط از خود برد (Orange Pi) یا سرور لوکال بتونه وصل بشه (host='localhost') کسی از بیرون نتونه مستقیم پورت دیتابیس رو باز کنه و وصل بشه. هر چیزی که نیاز نداره، قطع بشه.

مثلاً اگر برنامه فقط لاغ می‌نویسه → فقط INSERT

اگر فقط داده می‌خونه → فقط SELECT

اگر هم می‌نویسه و هم می‌خونه → INSERT + SELECT

¶ به این ترتیب، حتی اگر کسی برنامه رو هک کنه، نمی‌تونه کل دیتابیس رو نابود کنه یا اطلاعات حساس جای دیگه رو ببینه.

یعنی ایجاد کاربر دیتابیس با دسترسی محدود، فقط روی localhost، و فقط به همون جدول/عملیاتی که لازم داره.

3- توضیح کدها:

detect_audio_yamnet_db_ML:

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

```

import sounddevice as sd
import numpy as np
import tensorflow as tf
from scipy.signal import resample_poly
from fractions import Fraction
import time, random, math

import params as yamnet_params
import yamnet as yamnet_model

# ===== DB (PyMySQL) =====
ENABLE_DB = True
if ENABLE_DB:
    import pymysql
    DB = dict(host='localhost', user='iotapp', password='ChangeMe_Strong!',
              database='iotdash', charset='utf8mb4', autocommit=True)
def db_connect():
    return pymysql.connect(**DB)
db_conn = db_connect()
db_cur = db_conn.cursor()

def safe_insert(label, confidence, timestamp, node_id='Ali-Node', extra=None):
    if not ENABLE_DB:
        return
    global db_conn, db_cur
    sql = ('INSERT INTO audio_events(label, confidence, ts, node_id, raw_json) '

```

```

'VALUES(%s,%s,%s,%s,%s)'

raw = {"model":"YAMNet","extra":extra or {}}

args = (label, round(float(confidence)*100, 2), timestamp, node_id, str(raw))

try:
    db_cur.execute(sql, args)

except Exception as e:
    print("DB error:", e)

try: db_conn.close()

except: pass

try:
    db_conn = db_connect()

    db_cur = db_conn.cursor()

    db_cur.execute(sql, args)

except Exception as e2:
    print("DB reconnect failed:", e2)

```

```

# ---- انتخاب دیوایس ورودی ----

USB_INDEX = 2

sd.default.device = (USB_INDEX, None)

dev = sd.query_devices(USB_INDEX)

device_sr = int(dev['default_samplerate'])

print("Using input device:", dev)

print("Device samplerate:", device_sr)

```

```

# ---- مدل و کلاس‌ها ----

class_map_path = 'yamnet_class_map.csv'

```

```

params = yamnet_params.Params()

TARGET_SR = int(params.sample_rate)      # YAMNet expects 16000 Hz

yamnet = yamnet_model.yamnet_frames_model(params)

yamnet.load_weights('yamnet.h5')

CLASS_NAMES = yamnet_model.class_names(class_map_path)

# ----- ML-ADV: Post-processing helpers -----
def softmax_entropy(p):

    # p: 1D array of class probabilities (sums to 1)

    eps = 1e-9

    return -float(np.sum(p * np.log(p + eps)))

def snr_db(x):

    # crude SNR: signal power vs. a robust noise floor estimate

    x = np.asarray(x, dtype=np.float32)

    if x.size == 0:

        return -100.0

    power = np.mean(x**2) + 1e-12

    noise_floor = np.percentile((x - np.mean(x))**2, 10) + 1e-12

    return 10.0 * math.log10(power / noise_floor)

class TemporalSmoother:

    """Exponential moving average over frame-level scores."""

    def __init__(self, n_classes, alpha=0.6):

        self.alpha = float(alpha)

        self.state = np.zeros(n_classes, dtype=np.float32)

```

```

self.init = False

def update(self, scores_vec):
    if not self.init:
        self.state = scores_vec.astype(np.float32)
        self.init = True
    else:
        self.state = self.alpha * scores_vec + (1.0 - self.alpha) * self.state
    return self.state


class EventDetector:
    """
    Hysteresis ON/OFF + minimum duration + class whitelist.

    - on_th: threshold for entering event state
    - off_th: threshold for leaving event state
    - min_dur_s: minimum duration to confirm an event
    """

    def __init__(self, fps, on_th=0.38, off_th=0.22, min_dur_s=0.9,
                 allowed_classes=None, ban_background=True):
        self.fps = float(fps)
        self.on_th = float(on_th)
        self.off_th = float(off_th)
        self.min_frames = int(round(min_dur_s * fps))
        self.allowed = set(allowed_classes) if allowed_classes else None
        self.ban_background = ban_background
        self._active = False
        self._count = 0

```

```

self._current_cls = None

# A few YAMNet "background-ish" classes to ignore

self._bg = {
    "Silence", "Rustle", "Hum", "Noise", "Static", "Sine wave",
    "Inside, small room", "Inside, large room or hall"
}

def _is_valid_class(self, cls_name):
    if self.allowed and cls_name not in self.allowed:
        return False
    if self.ban_background and cls_name in self._bg:
        return False
    return True

def step(self, probs):
    """
    probs: 1D array of class probabilities (smoothed)
    Returns: (event_emitted: bool, event_label: str|None, event_conf: float|None,
    top3:list)
    """

    top_idx = int(np.argmax(probs))
    top_p = float(probs[top_idx])
    label = CLASS_NAMES[top_idx]

    # top-3 for logging
    top3_idx = list(np.argsort(probs)[-3:][::-1])

```

```

top3 = [(CLASS_NAMES[i], float(probs[i])) for i in top3_idx]

if not self._is_valid_class(label):
    # decay state if invalid class dominates
    if self._active and top_p < self.off_th:
        self._active, self._count, self._current_cls = False, 0, None
    return False, None, None, top3

if not self._active:
    if top_p >= self.on_th:
        if self._current_cls == label:
            self._count += 1
        else:
            self._current_cls = label
            self._count = 1
        if self._count >= self.min_frames:
            self._active = True
            self._count = 0
    return True, label, top_p, top3 # emit when we cross min duration

else:
    # staying active; if confidence drops below off_th we deactivate
    if top_p < self.off_th:
        self._active, self._count, self._current_cls = False, 0, None
    return False, None, None, top3

# ----- Streaming/Inference loop -----

```

```

DURATION_S = 3.0      # capture chunk seconds

RECORD_SR = device_sr

FRAME_RATE = 1.0 / 0.48  # YAMNet ~0.48s per patch -> ≈2.08 fps (approx)

SMOOTH_ALPHA = 0.65

ENTROPY_MAX = 2.6      # reject uncertain frames (lower is more confident)

SNR_MIN_DB = -5.0      # reject near-silence


smoother = TemporalSmoothen(n_classes=len(CLASS_NAMES),
alpha=SMOOTH_ALPHA)

detector = EventDetector(fps=2.05, on_th=0.40, off_th=0.25, min_dur_s=1.0,
allowed_classes=None, ban_background=True)

print("Recording...")

while True:

    # ---- ضبط ----

    audio = sd.rec(int(DURATION_S * RECORD_SR),
                    samplerate=RECORD_SR, channels=1, dtype='float32')

    sd.wait()

    mono = np.squeeze(audio)

    # SNR اولیه فیلتر
    if snr_db(mono) < SNR_MIN_DB:
        print("... skipped (low SNR / near-silence)")
        print("Recording...")
        continue

```

```

# resample 16 به k

if RECORD_SR != TARGET_SR:

    frac = Fraction(TARGET_SR, RECORD_SR).limit_denominator()

    up, down = frac.numerator, frac.denominator

    mono = resample_poly(mono, up, down).astype(np.float32)

# جرایی مدل

waveform = tf.convert_to_tensor(mono, dtype=tf.float32)

scores, embeddings, spectrogram = yamnet(waveform) # [frames, n_class]

frame_probs = tf.nn.softmax(scores, axis=-1).numpy() # تبدیل به احتمال()

# نرم سازی زمانی و میانگین

probs_mean = np.mean(frame_probs, axis=0).astype(np.float32)

probs_smooth = smoother.update(probs_mean)

# فیلتر عدم قطعیت با انتروپی

H = softmax_entropy(probs_smooth)

if H > ENTROPY_MAX:

    print(f"... skipped (uncertain, entropy={H:.2f})")

    print("Recording...")

    continue

# تشخیص رخداد با هیسترزیس/حداقل مدت

emitted, label, conf, top3 = detector.step(probs_smooth)

```

```

# نمایش top-3 (اختیاری)

t3txt = ", ".join([f"{n}:{p:.2f}" for n,p in top3])

if emitted:

    timestamp = time.strftime('%Y-%m-%d %H:%M:%S')

    # اگر خواستی از confidence واقعی مدل استفاده کنی، همین است؛
    # در غیر این صورت می توانی مثل نسخه‌ی قبلی دمو، کمی نویز اضافه کنی

    demo_conf = float(conf) * random.uniform(0.95, 1.05)
    demo_conf = max(0.0, min(1.0, demo_conf))

    print("\n🔊 Audio Event Detected")
    print(f"Type of audio event : {label}")
    print(f"Model's confidence level : {demo_conf:.0%}")
    print(f"Top3 : {t3txt}")
    print(f"Timestamp of the event : {timestamp}")
    print(f"Node or sender identifier : Ali Asadi\n")

    extra = {"top3": top3, "entropy": H}
    safe_insert(label, demo_conf, timestamp, extra=extra)

else:
    print(f"... no event (top3: {t3txt})")

print("Recording...")

```

توضیح :

این اسکریپت هر ۳ ثانیه یک بافر صدا از میکروفون USB می‌گیرد، آن را (در صورت نیاز) به ۱۶ kHz رسماًپل می‌کند، با مدل احتمال کلاس‌های صوتی را می‌گیرد، بعد با یک منطق ML-ADV نرم‌سازی زمانی + هیسترزیس + حداقل‌مدت + فیلتر انتروپی + فیلتر SNR فقط رخدادهای معتبر را تأیید و در جدول audio_events دیتابیس درج می‌کند.

وابستگی‌ها و ورودی‌ها

: sounddevice, numpy, tensorflow, scipy, pymysql پکیج‌ها

فایل‌های مدل/کلاس‌ها کنار اسکریپت yamnet.py: ماثول‌های yamnet.h5, yamnet_class_map.csv، و params.py

میکروفون USB با اندیس ۲ USB_INDEX = قابل تغییر

label, confidence, ts, node_id, MySQL/MariaDB دیتابیس و جدول audio_events باید ستون‌های raw_json

اتصال به دیتابیس

بخش safe_insert: وتابع ENABLE_DB

با pymysql وصل می‌شود.

رویداد را با درصد اطمینان (top3/entropy) و raw_json کمکی (confidence*100) درج می‌کند.

اگر خطا شد، auto-reconnect می‌کند و دوباره تلاش می‌کند.

دربیافت صوت و آماده‌سازی

۱. انتخاب دیوايس:

USB_INDEX = 2 و device_sr گرفتن.

۲. ضبط بافر:

DURATION_S = 3.0s. با: sd.rec(int(DURATION_S * RECORD_SR)) هر حلقه

3. فیلتر SNR سکوت/نویز

تابع `snr_db(mono)` نسبت سیگنال به نویز تقریبی را می‌سنجد.

اگر کمتر از $\text{SNR_MIN_DB} = -5.0$ باشد، این بافر **skip** می‌شود (برای کاهش پردازش بیهوده).

4. Resample

اگر نرخ ضبط با نرخ مورد نیاز مدل 16kHz متفاوت بود، با `resample_poly` به 16kHz تبدیل می‌کند.

(Inference) استنتاج مدل

: `tf.convert_to_tensor(mono)` تبدیل به تنسور

اجرای YAMNet خروجی `scores` بر حسب فریم‌ها و کلاس‌ها

اعمال `softmax → frame_probs` احتمال برای هر کلاس در هر فریم

: `probs_mean` میانگین روی فریم‌ها

نرم‌سازی زمانی با `EMA (TemporalSmoothening) → probs_smooth`

کاهش نوسان لحظه‌ای

(Entropy Gate) فیلتر عدم قطعیت

: `softmax_entropy(probs_smooth)` محاسبه‌ی آنتروپی توزیع احتمال

اگر آنتروپی از $\text{ENTROPY_MAX} = 2.6$ بیشتر بود، یعنی مدل خیلی نامطمئن است. **skip**.

(EventDetector) تشخیص رخداد پیشرفتہ

منطق تشخیص در کلاس `EventDetector:`

کلاس‌های پس زمینه مثل "Hum", "Rustle", "Noise" و ... نادیده گرفته می‌شوند.

: هیسترزیس

آستانه‌ی ورود `on_th = 0.40`

آستانه‌ی خروج `off_th = 0.25`

کمک می‌کند خروجی پایدارتر باشد و پرش نکند.

حداقل مدت ۰.۱۰۵fps باید پشت سر هم بالای آستانه بماند تا رخداد تأیید شود.

اگر شرایط برقرار شد، بر می گرداند (emitted=True, label, conf, top3)

ثبت و چاپ

اگر رخداد تأیید شد:

timestamp = now

چاپ اطلاعات رخداد بر حسب، اطمینان، top3

درج در DB با safe_insert همراه {top3, entropy} extra =

اگر رخداد تأیید نشد:

فقط top3 چاپ می شود.

نکته: مقدار demo_conf را کمی $\pm 5\%$ نویز می دهد؛ اگر دمو نمی خواهی، مستقیم conf را ذخیره کن.

پارامترهای قابل تبیون

DURATION_S, USB_INDEX ضبط

آستانه ها on_th, off_th, min_dur_s

نرم سازی SMOOTH_ALPHA

عدم قطعیت ENTROPY_MAX: سخت گیرتر کوچکتر

سکوت / نویز SNR_MIN_DB: سخت گیرتر بزرگتر

«Dog bark»: allowed_classes برای تنظیم ban_background=True کلاس ها «Speech», ...

خطاهای رایج و چک لیست

pip3 install apt install ... یا **ModuleNotFoundError** پکیج ها را نصب کنیم

YAMNet files: yamnet.h5, yamnet_class_map.csv, params.py, yamnet.py
درست باشند.

Device SR mismatch ریسمپل انجام می‌شود؛ اگر خطا داریم، بررسی کنیم میکروفون درست انتخاب شده باشد (sd.query_devices()).

DB insert error یوزر/پسورد/دیتابیس/جدول را چک باید کنیم؛ دسترسی iotapp و اسکیمای جدول.

۱ هدر فایل و ایمپورت‌ها

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

شبینگ: به سیستم می‌گوید با Python3 اجرا شود.

انکودینگ: متن فایل (کامنت‌ها/رشته‌ها) UTF-8 است.

```
import sounddevice as sd
```

```
import numpy as np
```

```
import tensorflow as tf
```

```
from scipy.signal import resample_poly
```

```
from fractions import Fraction
```

```
import time, random, math
```

sounddevice ضبط صدا از کارت صوتی/میکروفون.

numpy آرایه‌ها و محاسبات عددی.

tensorflow جرای مدل YAMNet.

resample_poly ریسمپل کردن سیگنال به 16 kHz.

Fraction یافتن نسبت کسری ساده برای up/down در ریسمپل.

math/random/time ریزمان، نویز دمو برای کانفیدنس، محاسبات لگاریتم. ... /

```
import params as yamnet_params
```

```
import yamnet as yamnet_model
```

YAMNet کمکی ماذول‌های هایپرپارامترها و تعریف خود مدل/کلاس‌ها

۲ اتصال دیتابیس و تابع درج امن

```

# ===== DB (PyMySQL) =====
ENABLE_DB = True
if ENABLE_DB:
    import pymysql
    DB = dict(host='localhost', user='iotapp', password='ChangeMe_Strong!',
              database='iotdash', charset='utf8mb4', autocommit=True)
def db_connect():
    return pymysql.connect(**DB)
db_conn = db_connect()
db_cur = db_conn.cursor()

```

اگر ENABLE_DB=True باشد:

pymysql را ایمپورت می‌کند.

پیکربندی اتصال را در دیکشنری DB می‌گذارد هاست/یوزر/پس و DB utf8mb4 و

تابع db_connect() برای ساخت اتصال.

در لحظه‌ی شروع، یک اتصال (db_conn) و کرسر (db_cur) می‌سازد.

```

def safe_insert(label, confidence, timestamp, node_id='Ali-Node', extra=None)

    if not ENABLE_DB
        return

    global db_conn, db_cur
    sql = ('INSERT INTO audio_events(label, confidence, ts, node_id, raw_json) '
           'VALUES(%s,%s,%s,%s,%s)')
    raw = {"model":"YAMNet","extra":extra or {}}
    args = (label, round(float(confidence)*100, 2), timestamp, node_id, str(raw))

    try:
        db_cur.execute(sql, args)

```

```

except Exception as e:
    print("DB error:", e)
try: db_conn.close()
except: pass
try:
    db_conn = db_connect()
    db_cur = db_conn.cursor()
    db_cur.execute(sql, args)
except Exception as e2:
    print("DB reconnect failed:", e2)

```

اگر DB غیرفعال است، هیچ کاری نکن.

درج در جدول SQL audio_events با ۵ ستون (label/confidence/ts/node_id/raw_json).

raw_json دیکشنری ساده از اطلاعات اضافی (top3/entropy) که به رشته تبدیل شده.

confidence به درصد ذخیره می‌شود (100× و رُند دو رقم).

اجرای execute()؛ اگر خطأ شد:

اتصال را می‌بندد (اگر باز بود) و دوباره وصل می‌شود و یک بار دیگر تلاش می‌کند.

اگر باز هم خطأ شد، پیام خطای دوم چاپ می‌شود.

3 انتخاب و آماده‌سازی دستگاه صوتی

```

# ----: انتخاب دیوایس ورودی ----
USB_INDEX = 2
sd.default.device = (USB_INDEX, None)
dev = sd.query_devices(USB_INDEX)
device_sr = int(dev['default_samplerate'])
print("Using input device:", dev)

```

```

print("Device samplerate:", device_sr)

sd.query_devices()).
اندیس کارت صدای ورودی شما (قابل تغییر با USB_INDEX=2
ورودی، خروجی خروجی را None گذاشتید.

sd.default.device=
اطلاعات دیوایس ۲ را می‌گیرد (برای چاپ اطمینان).
query_devices(2)
نرخ نمونه‌برداری پیش‌فرض همین دیوایس (مثلًا 44100 یا 48000).

device_sr
برای دیباگ.
دو print

```

4 بارگذاری مدل YAMNet و نگاشت کلاس‌ها

```

class_map_path = 'yamnet_class_map.csv'

params = yamnet_params.Params()

TARGET_SR = int(params.sample_rate)      # YAMNet expects 16000 Hz

yamnet = yamnet_model.yamnet_frames_model(params)

yamnet.load_weights('yamnet.h5')

CLASS_NAMES = yamnet_model.class_names(class_map_path)

CSV کلاس‌ها.
مسیر

```

Params() حاوی sample_rate=16000.

TARGET_SR=16000 برای این مدل.

ساخت خود مدل فریم-محور (yamnet_frames_model).

بارگذاری وزن‌های آموزش‌دیده از yamnet.h5

گرفتن لیست نام کلاس‌ها از CSV

5 توابع کمکی پس‌پردازش (ML-ADV)

```

def softmax_entropy(p
    eps = 1e-9
    return -float(np.sum(p * np.log(p + eps)))

```

آنتروپی توزیع احتمال p را حساب می‌کند (هرچه بزرگتر → عدم قطعیت بیشتر).

```

def snr_db(x):
    x = np.asarray(x, dtype=np.float32)
    if x.size == 0:
        return -100.0
    power = np.mean(x**2) + 1e-12
    noise_floor = np.percentile((x - np.mean(x))**2, 10) + 1e-12
    return 10.0 * math.log10(power / noise_floor)

```

برآوردهای خیلی ساده از **SNR**

توان کل سیگنال (power) نسبت به یک کف‌نویز مقاوم 10th percentile از توان سیگنال

خروجی دسیبل. اگر کم باشد، یعنی سکوت/پس‌زمینه‌ی کم قدرت \Rightarrow بهتر است skip.

```

class TemporalSmoother:
    """Exponential moving average over frame-level scores."""

    def __init__(self, n_classes, alpha=0.6):
        self.alpha = float(alpha)
        self.state = np.zeros(n_classes, dtype=np.float32)
        self.init = False

    def update(self, scores_vec):
        if not self.init:
            self.state = scores_vec.astype(np.float32)
            self.init = True
        else:
            self.state = self.alpha * scores_vec + (1.0 - self.alpha) * self.state
        return self.state

```

EMA روى بردار احتمال کلاس‌ها:

بار اول: state = ورودی.

بعد از آن $\alpha \cdot \text{state} \leftarrow \text{ورودی state} + (1-\alpha) \cdot \text{قبای}.$

هدف: نرم سازی زمانی و پایدار تر کردن تصمیم.

class EventDetector:

"""

Hysteresis ON/OFF + minimum duration + class whitelist.

- on_th: threshold for entering event state
- off_th: threshold for leaving event state
- min_dur_s: minimum duration to confirm an event

"""

def __init__(self, fps, on_th=0.38, off_th=0.22, min_dur_s=0.9,

allowed_classes=None, ban_background=True):

self.fps = float(fps)

self.on_th = float(on_th)

self.off_th = float(off_th)

self.min_frames = int(round(min_dur_s * fps))

self.allowed = set(allowed_classes) if allowed_classes else None

self.ban_background = ban_background

self._active = False

self._count = 0

self._current_cls = None

A few YAMNet “background-ish” classes to ignore

self._bg = {

“Silence”, “Rustle”, “Hum”, “Noise”, “Static”, “Sine wave”,

“Inside, small room”, “Inside, large room or hall”

}

تشخیص رویداد با هیسترزیس و حداقل مدت:

نرخ تصمیم تقریباً ~ 2 فریم بر ثانیه برای YAMNet fps

آستانه‌های ورود/خروج : on_th / off_th.

باید این تعداد فریم پیاپی بالای آستانه باشد تا رخداد تأیید شود. min_frames

اگر لیست بدهید، فقط همان کلاس‌ها مجازند. allowed

کلاس‌های پس‌زمینه لیست bg_background=True حذف می‌شوند.

متغیرهای وضعیت: فعال/غیرفعال، شمارنده، کلاس جاری.

```
def _is_valid_class(self, cls_name):
    if self.allowed and cls_name not in self.allowed:
        return False
    if self.ban_background and cls_name in self._bg:
        return False
    return True

    چک می‌کند کلاس فعلی برای اعلان رویداد مجاز هست یا نه بر اساس banlist, whitelist

def step(self, probs):
    """
    probs: 1D array of class probabilities (smoothed)
    Returns: (event_emitted: bool, event_label: str|None, event_conf: float|None,
    top3:list)
    """

    top_idx = int(np.argmax(probs))
    top_p = float(probs[top_idx])
    label = CLASS_NAMES[top_idx]
    # top-3 for logging
    top3_idx = list(np.argsort(probs)[-3:][::-1])
```

```
top3 = [(CLASS_NAMES[i], float(probs[i])) for i in top3_idx]
```

از بردار احتمال، کلاس غالب (label) و احتمال آن (top_p) را می‌گیرد.

سه کلاس برتر (top3) را برای گزارش/ثبت می‌سازد.

```
if not self._is_valid_class(label):
```

```
    # decay state if invalid class dominates
```

```
    if self._active and top_p < self.off_th:
```

```
        self._active, self._count, self._current_cls = False, 0, None
```

```
    return False, None, None, top3
```

اگر کلاس غالب معتبر نیست:

اگر در حالت فعال هستیم و اطمینان زیر آستانه‌ی خروج است → غیر فعال شو.

رویداد صادر نمی‌کند.

```
if not self._active:
```

```
    if top_p >= self.on_th:
```

```
        if self._current_cls == label:
```

```
            self._count += 1
```

```
        else:
```

```
            self._current_cls = label
```

```
            self._count = 1
```

```
        if self._count >= self.min_frames:
```

```
            self._active = True
```

```
            self._count = 0
```

```
    return True, label, top_p, top3 # emit when we cross min duration
```

اگر در حالت غیرفعال هستیم:

وقتی احتمال از آستانه ورود گذشت، شمارش فریم‌های پیاپی همان کلاس را افزایش بده.

اگر به حداقل مدت رسید → حالت فعال و رویداد صادر کن (emitted=True).

```

else:
    # staying active; if confidence drops below off_th we deactivate
    if top_p < self.off_th:
        self._active, self._count, self._current_cls = False, 0, None
return False, None, None, top3

```

اگر در حالت فعال هستیم:

وقتی احتمال زیر آستانه خروج بباید → غیر فعال شو.

در غیر این صورت، رویدادی صادر نمی شود.

6 پارامترهای حلقه‌ی استریم و ساخت آبجکت‌ها

```

DURATION_S = 3.0      # capture chunk seconds
RECORD_SR = device_sr
FRAME_RATE = 1.0 / 0.48  # YAMNet ~0.48s per patch -> ≈2.08 fps (approx)
SMOOTH_ALPHA = 0.65
ENTROPY_MAX = 2.6      # reject uncertain frames (lower is more confident)
SNR_MIN_DB = -5.0      # reject near-silence

```

طول هر بافر ضبط: ۳ ثانیه.

نرخ ضبط بر اساس دیوايس.

نرخ فریم YAMNet ≈ 2.08fps تقریبی، برای محاسبه‌ی min_dur به فریم

آلفای EMA، سقف آنتروپی مجاز بزرگ‌تر از این = عدم قطعیت زیاد skip

حداقل SNR بالاتر → سخت‌گیرتر

```

smoother = TemporalSmoothen(n_classes=len(CLASS_NAMES),
alpha=SMOOTH_ALPHA)

detector = EventDetector(fps=2.05, on_th=0.40, off_th=0.25, min_dur_s=1.0,
allowed_classes=None, ban_background=True)

```

ساخت EMA برای تعداد کلاس‌ها با $a=0.65$

ساخت تشخیص دهنده با $\text{fps} \approx 2.05$ ، آستانه‌ها $0.25/0.40$ ، حداقل مدت 1.0 پس زمینه.

```
print("Recording...")
```

صرفاً لاغ.

7 حلقه‌ی بی‌نهایت: ضبط → استنتاج → پس‌بردارش → درج

while True:

حلقه دائمی.

ضبط و آماده‌سازی:

```
audio = sd.rec(int(DURATION_S * RECORD_SR),  
    samplerate=RECORD_SR, channels=1, dtype='float32')  
  
sd.wait()  
  
mono = np.squeeze(audio)  
  
float32 با تعداد نمونه‌های DURATION_S * RECORD_SR sd.rec  
  
sd.wait() تا ضبط تمام شود.  
  
np.squeeze تبدیل به آرایه 1 بعدی.
```

SNR فیلتر

```
if snr_db(mono) < SNR_MIN_DB:  
    print("... skipped (low SNR / near-silence)")  
    print("Recording...")  
    continue
```

اگر SNR کمتر از -5 dB بود → این بافر را نادیده بگیر.

k16 ریسمپل به

```
if RECORD_SR != TARGET_SR:  
    frac = Fraction(TARGET_SR, RECORD_SR).limit_denominator()  
    up, down = frac.numerator, frac.denominator
```

```
mono = resample_poly(mono, up, down).astype(np.float32)
```

اگر نمونهبرداری کارت صوتی 16 کنیست:

نسبت کسری 16 را به شکل k /record_sr up/down رساده می‌کند.

با resample_poly ریسمپل می‌کند و float32 می‌گیرد.

استنتاج مدل:

```
waveform = tf.convert_to_tensor(mono, dtype=tf.float32)
```

```
scores, embeddings, spectrogram = yamnet(waveform) # [frames, n_class]
```

```
frame_probs = tf.nn.softmax(scores, axis=-1).numpy() # تبدیل به احتمال
```

سیگنال را تنسور می‌کند.

اجرای YAMNet \rightarrow scores روی فریم‌ها/کلاس‌ها.

روی محور کلاس‌ها برای تبدیل به احتمال softmax (numpy).

میانگین‌گیری و EMA

```
probs_mean = np.mean(frame_probs, axis=0).astype(np.float32)
```

```
probs_smooth = smoother.update(probs_mean)
```

روی فریم‌ها میانگین می‌گیرد (یک بردار احتمال برای کل بافر).

سپس EMA را اعمال می‌کند تا نرم شود.

(Entropy عدم قطعیت فیلتر)

```
H = softmax_entropy(probs_smooth)
```

```
if H > ENTROPY_MAX:
```

```
    print(f"... skipped (uncertain, entropy={H:.2f})")
```

```
    print("Recording...")
```

```
    continue
```

اگر آنتروپی زیاد باشد (پهن بودن توزیع احتمال \rightarrow مدل نامطمئن) \rightarrow رد.

تشخیص رخداد (هیسترزیس/حداقل مدت):

```
emitted, label, conf, top3 = detector.step(probs_smooth)
```

```
t3txt = ", ".join([f"{n}:{p:.2f}" for n,p in top3])
```

step تصمیم می‌گیرد آیا رویداد صادر شود یا نه، و برچسب اعتماد top3 را بر می‌گرداند.

رشته‌ی نمایشی top3 را می‌سازد.

اگر رویداد تأیید شد:

```
if emitted:
```

```
    timestamp = time.strftime('%Y-%m-%d %H:%M:%S')
```

```
    demo_conf = float(conf) * random.uniform(0.95, 1.05)
```

```
    demo_conf = max(0.0, min(1.0, demo_conf))
```

زمان فعلی.

برای نمایش دمو می‌توانید حذف کنید و خود conf را ذخیره کنید

کلمپ به [0,1].

```
print("\n🔊 Audio Event Detected")
```

```
print(f"Type of audio event : {label}")
```

```
print(f"Model's confidence level : {demo_conf:.0%}")
```

```
print(f"Top3 : {t3txt}")
```

```
print(f"Timestamp of the event : {timestamp}")
```

```
print(f"Node or sender identifier : Ali Asadi\n")
```

لاغ کاربرپسند.

```
extra = {"top3": top3, "entropy": H}
```

```
safe_insert(label, demo_conf, timestamp, extra=extra)
```

برای ذخیره‌ی top3 و آنتروپی extra ساخت.

درج در دیتابیس با تابع آمن.

اگر رویداد نبود:

```
else:  
    print(f"... no event (top3: {t3txt})")
```

فقط گزارش می‌دهد که رخداد تأیید نشد.

پایان سیکل:

```
print("Recording...")
```

اعلام شروع سیکل بعدی.

```
detect_audio_yamnet_db:
```

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

```
import sounddevice as sd  
import numpy as np  
import tensorflow as tf  
from scipy.signal import resample_poly  
from fractions import Fraction  
import time
```

```
import params as yamnet_params  
import yamnet as yamnet_model  
اگر نبود مشکلی نیست؛ توی نسخه تو بود #  
import features
```

```
# ===== DB (PyMySQL) =====
```

```
ENABLE_DB = True
```

```
if ENABLE_DB:
```

```
    import pymysql
```

```

DB = dict(host='localhost', user='iotapp', password='ChangeMe_Strong!',
          database='iotdash', charset='utf8mb4', autocommit=True)

def db_connect():
    return pymysql.connect(**DB)

db_conn = db_connect()
db_cur = db_conn.cursor()

# ---- انتخاب دیوايس ورودی ----
USB_INDEX = 2
sd.default.device = (USB_INDEX, None)
dev = sd.query_devices(USB_INDEX)
device_sr = int(dev['default_samplerate'])
print("Using input device:", dev)
print("Device samplerate:", device_sr)

# ---- مدل و کلاسها ----
class_map_path = 'yamnet_class_map.csv'
params = yamnet_params.Params()
target_sr = int(params.sample_rate)      # YAMNet = 16000
yamnet = yamnet_model.yamnet_frames_model(params)
yamnet.load_weights('yamnet.h5')
class_names = yamnet_model.class_names(class_map_path)

```

```

# ---- ضبط ----

duration = 10

record_sr = device_sr


print("Recording...")

def safe_insert(label, confidence, timestamp, node_id='Ali-Node'):

    if not ENABLE_DB:

        return

    global db_conn, db_cur

    sql = ('INSERT INTO audio_events(label, confidence, ts, node_id, raw_json) '
           "'VALUES(%s,%s,%s,%s,%s)')

    # ذخیره به صورت درصد # ذخیره به صورت درصد (0..100)

    args = (label, round(float(confidence) * 100, 2), timestamp, node_id,
    {"model":"YAMNet"})

    try:

        db_cur.execute(sql, args)

    except Exception as e:

        print("DB error:", e)

        # یکباره reconnect تلاش برای

    try:

        db_conn.close()

    except:

        pass

    try:

        db_conn = db_connect()

```

```

db_cur = db_conn.cursor()
db_cur.execute(sql, args)

except Exception as e2:
    print("DB reconnect failed:", e2)

while True:
    # ضبط
    audio = sd.rec(int(duration * record_sr),
                    samplerate=record_sr, channels=1, dtype='float32')
    sd.wait()

    mono_audio = np.squeeze(audio)

    # resample 16 به در صورت نیاز
    if record_sr != target_sr:
        frac = Fraction(target_sr, record_sr).limit_denominator()
        up, down = frac.numerator, frac.denominator
        mono_audio = resample_poly(mono_audio, up, down).astype(np.float32)

    # اجرای مدل
    waveform = tf.convert_to_tensor(mono_audio, dtype=tf.float32)
    scores, embeddings, spectrogram = yamnet(waveform) # scores: [frames,
    num_classes] در بازه [0,1]

    # میانگین گیری روی فریم‌ها و انتخاب برترین کلاس
    mean_scores = tf.reduce_mean(scores, axis=0).numpy() # -> ndarray
    [num_classes]

```

```

top_index = int(np.argmax(mean_scores))

label = class_names[top_index]

confidence = float(mean_scores[top_index])

timestamp = time.strftime('%Y-%m-%d %H:%M:%S')

if label in ["Rustle", "Hum"]:

    print(f"❌ No meaningful audio event detected (label={label},
conf={confidence:.2%})")

else:

    print("\n🔊 Audio Event Detected")

    print(f"Type of audio event : {label}")

    print(f"Model's confidence level : {confidence:.0%}")

    print(f"Timestamp of the event : {timestamp}")

    print(f"Node or sender identifier : Ali Asadi\n")

# همان ۱..۰ را می‌دهیم؛ داخل تابع به درصد تبدیل می‌شود) درج در DB
safe_insert(label, confidence, timestamp)

print("Recording...")

```

توضیح :

شروع و وارد کردن ماثول‌ها

اول برنامه کتابخانه‌های لازم رو می‌آرده:

برای ضبط صدا از میکروفون **sounddevice**.

برای محاسبات عددی **numpy**.

چون مدل YAMNet روی اون اجرا می‌شه.

برای تغییر نرخ نمونه **scipy.signal.resample_poly**

برای محاسبه نسبت دقیق **fractions.Fraction**

برای گرفتن زمان و مهرزمان **time**

بعد هم فایل‌های مخصوص YAMNet (params و features) و در صورت نیاز yamnet لود می‌شن.

انتخاب میکروفون

برنامه مشخص می‌کنه که میکروفون ورودی کدومه اینجا ایندکس 2 بعد مشخصات دستگاه رو می‌گیره و نرخ نمونه پیش‌فرضش رو پرینت می‌کنه. مثلًا 44100 یا 48000 هرتز.

بارگذاری مدل YAMNet

سیر فایل کلاس‌ها (yamnet_class_map.csv) مشخص می‌شه.

پارامترها (از جمله نرخ نمونه 16000 هرتز) از فایل params خونده می‌شن.

مدل فریممحور YAMNet ساخته و وزن‌هاش از فایل yamnet.h5 بارگذاری می‌شن.

لیست نام کلاس‌ها از فایل CSV خونده می‌شه (بیش از ۵۰۰ کلاس صوتی مختلف).

آماده‌سازی ضبط

نرخ ضبط همون نرخ کارت صداست بعداً اگر با 16000 فرق داشت، resample می‌کنه

یک پیام "Recording..." چاپ می‌کنه.

تابع ثبت در دیتابیس (**safe_insert**)

این تابع اسم رویداد صوتی، میزان اطمینان (**confidence**) و زمان رو می‌گیره.
یه INSERT تو جدول **audio_events** می‌زنه.
اگه خطا پیش بیاد (مثلاً اتصال قطع باشد)، یه بار دیگه **reconnect** می‌کنه و دوباره سعی می‌کنه.

حلقه اصلی (بی‌نهایت)

از اینجا به بعد، برنامه پشت سر هم بسته‌های ۱۰ ثانیه‌ای پردازش می‌کنه:

۱. ضبط صدا:

۱۰ ثانیه از میکروفون می‌گیره، تک کاناله ذخیره می‌کنه.

۲. تغییر نرخ نمونه (اگر لازم بود):

اگر نرخ ضبط \neq ۱۶۰۰۰ باشد، **resample_poly** را دقیق به ۱۶۰۰۰ تبدیل می‌کنه.

۳. اجرای مدل **YAMNet**:

سیگнал به TensorFlow داده می‌شود.

خروجی سه چیزه:

scores احتمال هر کلاس برای هر فریم.

embeddings بردار ویژگی (استفاده نمی‌شود اینجا).

spectrogram طیف انرژی (اختیاری).

۴. میانگین‌گیری و انتخاب برترین کلاس:

روی تمام فریم‌ها میانگین می‌گیره \rightarrow یک بردار احتمال نهایی برای کل ۱۰ ثانیه.
بیشترین مقدار رو پیدا می‌کند و نام کلاس اون رو انتخاب می‌کند.

۵. ثبت در دیتابیس:

اگه صدا معنی‌دار باشد، اطلاعات رو به جدول **audio_events** می‌فرسته.

۶. شروع دوباره:

پیام "...Recording..." چاپ می‌شود و چرخه بعدی شروع می‌شود.

نتیجه

صدا ضبط → پردازش → تشخیص برچسب → ثبت در دیتابیس.
اینطوری می‌توانی تاریخچه رویدادهای صوتی محیط را به صورت خودکار در MySQL ذخیره کنی.

features.py:

```
"""Feature computation for YAMNet."""

import numpy as np
import tensorflow as tf

def waveform_to_log_mel_spectrogram_patches(waveform, params):
    """Compute log mel spectrogram patches of a 1-D waveform."""
    with tf.name_scope('log_mel_features'):
        # waveform has shape [<# samples>]

        # Convert waveform into spectrogram using a Short-Time Fourier Transform.
        # Note that tf.signal.stft() uses a periodic Hann window by default.

        window_length_samples = int(
            round(params.sample_rate * params.stft_window_seconds))

        hop_length_samples = int(
            round(params.sample_rate * params.stft_hop_seconds))

        fft_length = 2 ** int(np.ceil(np.log(window_length_samples) / np.log(2.0)))
```

```

num_spectrogram_bins = fft_length // 2 + 1

if params.tflite_compatible:

    magnitude_spectrogram = _tflite_stft_magnitude(
        signal=waveform,
        frame_length=window_length_samples,
        frame_step=hop_length_samples,
        fft_length=fft_length)

else:

    magnitude_spectrogram = tf.abs(tf.signal.stft(
        signals=waveform,
        frame_length=window_length_samples,
        frame_step=hop_length_samples,
        fft_length=fft_length))

# magnitude_spectrogram has shape [<# STFT frames>, num_spectrogram_bins]

# Convert spectrogram into log mel spectrogram.

linear_to_mel_weight_matrix = tf.signal.linear_to_mel_weight_matrix(
    num_mel_bands=params.mel_bands,
    num_spectrogram_bins=num_spectrogram_bins,
    sample_rate=params.sample_rate,
    lower_edge_hertz=params.mel_min_hz,
    upper_edge_hertz=params.mel_max_hz)

mel_spectrogram = tf.matmul(
    magnitude_spectrogram, linear_to_mel_weight_matrix)

log_mel_spectrogram = tf.math.log(mel_spectrogram + params.log_offset)

# log_mel_spectrogram has shape [<# STFT frames>, params.mel_bands]

# Frame spectrogram (shape [<# STFT frames>, params.mel_bands]) into patches
# (the input examples). Only complete frames are emitted, so if there is

```

```

# less than params.patch_window_seconds of waveform then nothing is emitted
# (to avoid this, zero-pad before processing).

spectrogram_hop_length_samples = int(
    round(params.sample_rate * params.stft_hop_seconds))

spectrogram_sample_rate = params.sample_rate / spectrogram_hop_length_samples

patch_window_length_samples = int(
    round(spectrogram_sample_rate * params.patch_window_seconds))

patch_hop_length_samples = int(
    round(spectrogram_sample_rate * params.patch_hop_seconds))

features = tf.signal.frame(
    signal=log_mel_spectrogram,
    frame_length=patch_window_length_samples,
    frame_step=patch_hop_length_samples,
    axis=0)

# features has shape [<# patches>, <# STFT frames in an patch>, params.mel_bands]

return log_mel_spectrogram, features

```

```

def pad_waveform(waveform, params):
    """Pads waveform with silence if needed to get an integral number of patches."""

    # In order to produce one patch of log mel spectrogram input to YAMNet, we
    # need at least one patch window length of waveform plus enough extra samples
    # to complete the final STFT analysis window.

    min_waveform_seconds = (
        params.patch_window_seconds +
        params.stft_window_seconds - params.stft_hop_seconds)

    min_num_samples = tf.cast(min_waveform_seconds * params.sample_rate, tf.int32)

    num_samples = tf.shape(waveform)[0]

```

```

num_padding_samples = tf.maximum(0, min_num_samples - num_samples)

# In addition, there might be enough waveform for one or more additional
# patches formed by hopping forward. If there are more samples than one patch,
# round up to an integral number of hops.

num_samples = tf.maximum(num_samples, min_num_samples)
num_samples_after_first_patch = num_samples - min_num_samples

hop_samples = tf.cast(params.patch_hop_seconds * params.sample_rate, tf.int32)
num_hops_after_first_patch = tf.cast(tf.math.ceil(
    tf.cast(num_samples_after_first_patch, tf.float32) /
    tf.cast(hop_samples, tf.float32)), tf.int32)

num_padding_samples += (
    hop_samples * num_hops_after_first_patch - num_samples_after_first_patch)

padded_waveform = tf.pad(waveform, [[0, num_padding_samples]],
    mode='CONSTANT', constant_values=0.0)

return padded_waveform


def _tflite_stft_magnitude(signal, frame_length, frame_step, fft_length):
    """TF-Lite-compatible version of tf.abs(tf.signal.stft())."""

def _hann_window():
    return tf.reshape(
        tf.constant(
            (0.5 - 0.5 * np.cos(2 * np.pi * np.arange(0, 1.0, 1.0 / frame_length)))
            .astype(np.float32),
            name='hann_window'), [1, frame_length])


def _dft_matrix(dft_length):

```

```

"""Calculate the full DFT matrix in NumPy."""

# See https://en.wikipedia.org/wiki/DFT_matrix

omega = (0 + 1j) * 2.0 * np.pi / float(dft_length)

# Don't include 1/sqrt(N) scaling, tf.signal.rfft doesn't apply it.

return np.exp(omega * np.outer(np.arange(dft_length), np.arange(dft_length)))

```



```

def _rdft(framed_signal, fft_length):

    """Implement real-input Discrete Fourier Transform by matmul."""

    # We are right-multiplying by the DFT matrix, and we are keeping only the
    # first half ("positive frequencies"). So discard the second half of rows,
    # but transpose the array for right-multiplication. The DFT matrix is
    # symmetric, so we could have done it more directly, but this reflects our
    # intention better.

    complex_dft_matrix_kept_values = _dft_matrix(fft_length)[:(

        fft_length // 2 + 1), :].transpose()

    real_dft_matrix = tf.constant(
        np.real(complex_dft_matrix_kept_values).astype(np.float32),
        name='real_dft_matrix')

    imag_dft_matrix = tf.constant(
        np.imag(complex_dft_matrix_kept_values).astype(np.float32),
        name='imaginary_dft_matrix')

    signal_frame_length = tf.shape(framed_signal)[-1]

    half_pad = (fft_length - signal_frame_length) // 2

    padded_frames = tf.pad(
        framed_signal,
        [
            # Don't add any padding in the frame dimension.
            [0, 0],
            # Pad before and after the signal within each frame.

```

```

[half_pad, fft_length - signal_frame_length - half_pad]

],
mode='CONSTANT',
constant_values=0.0)

real_stft = tf.matmul(padded_frames, real_dft_matrix)

imag_stft = tf.matmul(padded_frames, imag_dft_matrix)

return real_stft, imag_stft

def _complex_abs(real, imag):

    return tf.sqrt(tf.add(real * real, imag * imag))

framed_signal = tf.signal.frame(signal, frame_length, frame_step)

windowed_signal = framed_signal * _hann_window()

real_stft, imag_stft = _rdft(windowed_signal, fft_length)

stft_magnitude = _complex_abs(real_stft, imag_stft)

return stft_magnitude

```

این کد مسئول تبدیل سیگنال صوتی خام به ویژگی‌هایی است که مدل YAMNet می‌فهمد؛ یعنی

log-mel spectrogram patches

بخش اول: تابع اصلی استخراج ویژگی

def waveform_to_log_mel_spectrogram_patches(waveform, params):

این تابع سیگنال صوتی خام (waveform) را به log-mel spectrogram patch‌های (patch) تبدیل می‌کند که ورودی مدل YAMNet هستند.

گام 1: محاسبه طول پنجره و گام STFT

window_length_samples = int(round(params.sample_rate *
params.stft_window_seconds))

```
hop_length_samples = int(round(params.sample_rate * params.stft_hop_seconds))
```

FFT تعداد نمونه‌ها در هر پنجره‌ی window_length_samples •

فاصله‌ی hop_length_samples بین شروع پنجره‌ها. •

گام 2: محاسبه طول FFT

```
fft_length = 2 ** int(np.ceil(np.log(window_length_samples) / np.log(2.0)))
```

برای کارایی بهتر در window_length_samples برابر است با نزدیک‌ترین توان دو بزرگ‌تر یا مساوی از FFT طول FFT).

گام 3: محاسبه طیف دامنه (magnitude spectrogram)

```
if params.tflite_compatible:
```

```
    magnitude_spectrogram = _tflite_stft_magnitude(...)
```

```
else:
```

```
    magnitude_spectrogram = tf.abs(tf.signal.stft(...))
```

بسته به اینکه مدل قابل تبدیل به TFLite باشد یا نه:

از tf.signal.stft برای محاسبه طیف فوریه استفاده می‌شود. •

یا از نسخه‌ی دستی _tflite_stft_magnitude استفاده می‌شود که در پایین تعریف شده است. •

گام 4: تبدیل طیف به مقیاس mel

```
linear_to_mel_weight_matrix = tf.signal.linear_to_mel_weight_matrix(...)
```

```
mel_spectrogram = tf.matmul(magnitude_spectrogram, linear_to_mel_weight_matrix)
```

```
log_mel_spectrogram = tf.math.log(mel_spectrogram + params.log_offset)
```

ایک ماتریس وزن است که طیف را از مقیاس خطی به مقیاس mel تبدیل می‌کند.

سپس log گرفته می‌شود برای شباهت بیشتر به درک شنوایی انسان.

گام 5: تبدیل patch‌ها به log-mel spectrogram

```
spectrogram_sample_rate = params.sample_rate / spectrogram_hop_length_samples  
patch_window_length_samples = int(round(spectrogram_sample_rate *  
params.patch_window_seconds))  
patch_hop_length_samples = int(round(spectrogram_sample_rate *  
params.patch_hop_seconds))  
  
features = tf.signal.frame(signal=log_mel_spectrogram,  
                           frame_length=patch_window_length_samples,  
                           frame_step=patch_hop_length_samples,  
                           axis=0)
```

- `tf.signal.frame` را به پنجره‌های متوالی تقسیم می‌کند
- این `patch`‌ها همان چیزی هستند که به مدل YAMNet داده می‌شود.

خروجی این تابع:

```
return log_mel_spectrogram, features
```

pad_waveform:

```
def pad_waveform(waveform, params):
```

```
"""Pads waveform with silence if needed to get an integral number of patches."""
```

هدف این تابع:

اگر طول `waveform` (صدا) برای تولید حداقل یک `patch` کافی نباشد یا برای `patch`‌های کامل کافی نباشد، این تابع آن را با صفر (سکوت) `padding` می‌کند تا آماده‌ی پردازش باشد.

گام 1: محاسبه حداقل تعداد نمونه مورد نیاز

```

min_waveform_seconds = (
    params.patch_window_seconds +
    params.stft_window_seconds - params.stft_hop_seconds)

min_num_samples = tf.cast(min_waveform_seconds * params.sample_rate, tf.int32)

```

- برای داشتن یک **patch** کامل، باید حداقل به اندازه‌ی یک پنجره **patch** و یک پنجره FFT کم شده با **hop** داده داشته باشیم.

- این مقدار به تعداد نمونه‌ها تبدیل می‌شود.

2: بررسی کم بودن تعداد نمونه‌ها و محاسبه **padding**

```

num_samples = tf.shape(waveform)[0]

num_padding_samples = tf.maximum(0, min_num_samples - num_samples)

```

- اگر **waveform** کوتاه‌تر از مقدار حداقل باشد، تفاوت را محاسبه کرده و به صورت صفر **padding** می‌کند.

3: اگر **waveform** بلندتر باشد، **padding** را برای تعداد کامل **patch**‌ها تنظیم کن

```

num_samples = tf.maximum(num_samples, min_num_samples)

num_samples_after_first_patch = num_samples - min_num_samples

hop_samples = tf.cast(params.patch_hop_seconds * params.sample_rate, tf.int32)

num_hops_after_first_patch = tf.cast(tf.math.ceil(
    tf.cast(num_samples_after_first_patch, tf.float32) /
    tf.cast(hop_samples, tf.float32)), tf.int32)

```

- اگر بیشتر از یک **patch** می‌توان ساخت، بررسی می‌شود که برای تمام **hop**‌ها تعداد نمونه کافی هست یا نه.

- اگر نه، باز هم **padding** انجام می‌شود.

4: اعمال **padding** به انتهای **waveform**

```
num_padding_samples += (
```

```
hop_samples * num_hops_after_first_patch - num_samples_after_first_patch)
```

```
padded_waveform = tf.pad(waveform, [[0, num_padding_samples]],  
    mode='CONSTANT', constant_values=0.0)
```

باقیمانده لازم برای کامل شدن **patch** ها به مجموع **padding** اضافه می شود.

با مقدار صفر (سکوت) انجام می گیرد.

خروجی:

```
return padded_waveform
```

تابع **_tflite_stft_magnitude**

این تابع یک نسخه‌ی سازگار با **TFLite** از STFT (Short-Time Fourier Transform) است که خودش به صورت دستی:

• پنجره‌گذاری **Hann**

• محاسبه تبدیل فوریه‌ی گسسته **(DFT)**

• محاسبه اندازه طیف **(magnitude)**

را انجام می‌دهد. حالا خط به خط توضیح می‌دهم:

تعریف تابع:

```
def _tflite_stft_magnitude(signal, frame_length, frame_step, fft_length):
```

signal: سیگنال ورودی (waveform)

frame_length: طول هر فریم

frame_step: فاصله بین فریمهای (hop)

fft_length: طول تبدیل فوریه (frame_length) معمولاً نزدیکترین توان دو به (دو برابر frame_length)

زیرتابع _hann_window

```
def _hann_window():
    return tf.reshape(
        tf.constant(
            (0.5 - 0.5 * np.cos(2 * np.pi * np.arange(0, 1.0, 1.0 / frame_length)))
            .astype(np.float32),
            name='hann_window'), [1, frame_length])
```

• یک پنجره **Hann** را محاسبه می‌کند.

• باعث کاهش نشت طیفی (spectral leakage) می‌شود.

زیرتابع _dft_matrix

```
def _dft_matrix(dft_length):
    omega = (0 + 1j) * 2.0 * np.pi / float(dft_length)
    return np.exp(omega * np.outer(np.arange(dft_length), np.arange(dft_length)))
```

• یک ماتریس **DFT** کامل ایجاد می‌کند.

• این ماتریس برای محاسبه‌ی DFT به صورت ضرب ماتریسی استفاده می‌شود.

زیرتابع rdft_اعمال تبدیل فوریه‌ی واقعی با ماتریس DFT

```
def _rdft(framed_signal, fft_length):
```

1. ماتریس DFT را به دو قسمت **واقعی (real)** و **موهومی (imag)** تقسیم می‌کند.

2. سیگنال فریم‌بندی شده را صفرپد می‌کند (اگر لازم باشد).

3. هر فریم را ضرب ماتریسی با ماتریس DFT انجام می‌دهد.

زیرتابع _complex_abs

```
def _complex_abs(real, imag):
    return tf.sqrt(tf.add(real * real, imag * imag))
```

- اندازه عدد مختلط را محاسبه می کند:

$$|X| = \sqrt{(\text{Re}^2 + \text{Im}^2)}$$

اجرای کلی تابع:

```
framed_signal = tf.signal.frame(signal, frame_length, frame_step)

windowed_signal = framed_signal * _hann_window()

real_stft, imag_stft = _rdft(windowed_signal, fft_length)

stft_magnitude = _complex_abs(real_stft, imag_stft)

return stft_magnitude
```

1. فریم‌بندی سیگنال ورودی.

2. اعمال پنجره **Hann** روی هر فریم.

3. محاسبه **STFT DFT** واقعی

4. محاسبه اندازه طیف (**magnitude spectrum**) برای هر فریم.

خروجی: ماتریسی از مقادیر magnitude STFT با شکل تعداد فریم‌ها، تعداد فرکانس‌ها

حالا باید سراغ مهم‌ترین تابع این فایل:

تابع اصلی **:waveform_to_log_mel_spectrogram_patches**

```
def waveform_to_log_mel_spectrogram_patches(waveform, params):
```

"""Compute log mel spectrogram patches of a 1-D waveform."""

این تابع از یک سیگنال خام صوتی (waveform) سه چیز تولید می کند:

1. طیف **log-mel** ویژگی فشرده‌شده‌ی صدا

YAMNet بخش‌های جدا شده برای ورودی **patches** .2

و به صورت ضمنی **spectrogram** اولیه با **STFT** .3

گام 1: تنظیمات اولیه **STFT**

```
window_length_samples = int(round(params.sample_rate *  
params.stft_window_seconds))  
  
hop_length_samples = int(round(params.sample_rate * params.stft_hop_seconds))  
  
fft_length = 2 ** int(np.ceil(np.log(window_length_samples) / np.log(2.0)))  
  
num_spectrogram_bins = fft_length // 2 + 1
```

- محاسبه اندازه پنجره و گام (hop) بر اساس نرخ نمونه‌برداری.
- طول تبدیل فوریه (باید توان ۲ باشد).
- تعداد فرکانس‌ها بعد از **STFT** num_spectrogram_bins

گام 2: محاسبه **magnitude spectrogram**

```
if params.tflite_compatible:  
  
    magnitude_spectrogram = _tflite_stft_magnitude(...)  
  
else:  
  
    magnitude_spectrogram = tf.abs(tf.signal.stft(...))
```

- اگر **TFLite** استفاده شود، تابع دستی **tflite_stft_magnitude** اجرا می‌شود.
- در غیر این صورت از **TensorFlow** اصلی **STFT** استفاده می‌شود.

[frame, freq] خروجی

گام 3: تبدیل به طیف **mel**

```
linear_to_mel_weight_matrix = tf.signal.linear_to_mel_weight_matrix(...)
```

```
mel_spectrogram = tf.matmul(magnitude_spectrogram, linear_to_mel_weight_matrix)
```

- طیف خطی فرکانسی را به طیف Mel (شبیه به شنیدن انسان) تبدیل می‌کند.
- ماتریس نگاشت خطی به Mel ساخته و ضرب می‌شود.

گام 4: اعمال log

```
log_mel_spectrogram = tf.math.log(mel_spectrogram + params.log_offset)
```

- تبدیل به مقیاس لگاریتمی برای فشرده‌سازی دینامیک سیگنال.
- برای جلوگیری از $\log(0)$ است.

گام 5: تقسیم‌بندی patches (Framing)

```
spectrogram_sample_rate = params.sample_rate / hop_length_samples
```

```
patch_window_length_samples = int(round(spectrogram_sample_rate *  
params.patch_window_seconds))
```

```
patch_hop_length_samples = int(round(spectrogram_sample_rate *  
params.patch_hop_seconds))
```

```
features = tf.signal.frame(  
    signal=log_mel_spectrogram,  
    frame_length=patch_window_length_samples,  
    frame_step=patch_hop_length_samples,  
    axis=0)
```

- طیف log-mel به بخش‌های (patches) با طول مشخص تقسیم می‌شود.
- شکل نهایی [patch, frames_in_patch, mel_bands]

خروجی:

```
return log_mel_spectrogram, features
```

• **log_mel_spectrogram** : مکال طیف لگاریتمی در مقیاس Mel

• **features** : بخش‌های جدا شده آماده برای دادن به مدل YAMNet

این ساختار پایه‌ای ترین قدم برای استخراج ویژگی‌های صوتی در YAMNet هست.

: pad_waveform(waveform, params) تابع دوم

این تابع وظیفه دارد که اگر سیگنال خام صوتی (waveform) خیلی کوتاه باشد، آن را با سکوت (+) پد کند تا برای تقسیم‌بندی (framing) آماده شود.

چرا نیاز است؟

مدل YAMNet فقط روی بخش‌هایی (patches) از صدا کار می‌کند که به اندازه کافی طول دارند. اگر کمتر از حد لازم باشند، باید با + پر شوند تا patch کامل شود.

عملکرد خط به خط:

```
min_waveform_seconds = (
```

```
    params.patch_window_seconds +
```

```
    params.stft_window_seconds - params.stft_hop_seconds)
```

• حداقل زمانی که نیاز داریم برای ساختن یک patch کامل.

```
min_num_samples = tf.cast(min_waveform_seconds * params.sample_rate, tf.int32)
```

• تبدیل آن زمان به تعداد نمونه.(sample)

```
num_samples = tf.shape(waveform)[0]
```

```
num_padding_samples = tf.maximum(0, min_num_samples - num_samples)
```

• اگر طول سیگنال کمتر از حداقل است، تعداد نمونه‌هایی که باید اضافه شوند محاسبه می‌شود.

```
hop_samples = tf.cast(params.patch_hop_seconds * params.sample_rate, tf.int32)
```

...

```
num_padding_samples += (hop_samples * num_hops_after_first_patch -  
num_samples_after_first_patch)
```

- اگر بیشتر از یک patch لازم باشد، محاسبه می‌کند چند بار باید hop بزنیم و چقدر padding لازم است.

```
padded_waveform = tf.pad(waveform, [[0, num_padding_samples]], mode='CONSTANT',  
constant_values=0.0)
```

- پدینگ واقعی با صفر.

: `_tflite_stft_magnitude(...)`

این تابع یک جایگزین سبک و ساده برای `tf.signal.stft` استفاده می‌شود.

توضیح کامل آن را اگر خواستی می‌توانیم به صورت جداگانه بدیم، چون شامل:

- محاسبه دستی پنجره Hann
- تبديل فوريه گسسته با ماترييس DFT
- محاسبه اندازه (magnitude) از بخش موهومند و حقيقى

جمع‌بندی نهايى:

فایل `features.py` اينه که:

1. از waveform log-mel ويزگى های بگيره
2. اونا رو پد و تكه‌تكه کنه برای مدل YAMNet
3. نسخه‌ی سبک‌تر STFT برای TFLite هم پياده‌سازی کرده

yamnet.py:

""".Core model definition of YAMNet""""

```

import csv

import numpy as np

import tensorflow as tf

from tf_keras import Model, layers

import features as features_lib


:def _batch_norm(name, params)

:def _bn_layer(layer_input)

>return layers.BatchNormalization

, name=name

, center=params.batchnorm_center

, scale=params.batchnorm_scale

(epsilon=params.batchnorm_epsilon)(layer_input

return _bn_layer


:def _conv(name, kernel, stride, filters, params)

:def _conv_layer(layer_input)

, output = layers.Conv2D(name='{}/conv'.format(name

, filters=filters

, kernel_size=kernel

, strides=stride

, padding=params.conv_padding

```

```

,use_bias=False

(activation=None)(layer_input

output = _batch_norm('{}_conv/bn'.format(name), params)(output)

output = layers.ReLU(name='{}_relu'.format(name))(output)

return output

return _conv_layer

:def _separable_conv(name, kernel, stride, filters, params)

:def _separable_conv_layer(layer_input)

,output = layers.DepthwiseConv2D(name='{}_depthwise_conv'.format(name

,kernel_size=kernel

,strides=stride

,depth_multiplier=1

,padding=params.conv_padding

,use_bias=False

(activation=None)(layer_input

output = _batch_norm('{}_depthwise_conv/bn'.format(name), params)(output)

output = layers.ReLU(name='{}_depthwise_conv/relu'.format(name))(output)

,output = layers.Conv2D(name='{}_pointwise_conv'.format(name

,filters=filters

,(1,1)=kernel_size

,strides=1

,padding=params.conv_padding

,use_bias=False

(activation=None)(output

output = _batch_norm('{}_pointwise_conv/bn'.format(name), params)(output)

```

```

output = layers.ReLU(name='{}/pointwise_conv/relu'.format(name))(output)
return output

return _separable_conv_layer

] = YAMNET_LAYER_DEFS_
(layer_function, kernel, stride, num_filters)#
,(conv,      [3, 3], 2,  32_)
,(separable_conv, [3, 3], 1,  64_)
,(separable_conv, [3, 3], 2,  128_)
,(separable_conv, [3, 3], 1,  128_)
,(separable_conv, [3, 3], 2,  256_)
,(separable_conv, [3, 3], 1,  256_)
,(separable_conv, [3, 3], 2,  512_)
,(separable_conv, [3, 3], 1,  512_)
,(separable_conv, [3, 3], 1,  512_)
,(separable_conv, [3, 3], 1,  512_)
,(separable_conv, [3, 3], 2, 1024_)
(separable_conv, [3, 3], 1, 1024_)

[

```

```

:def yamnet(features, params)
"""Define the core YAMNet mode in Keras"""
>net = layers.Reshape

```

```
.(params.patch_frames, params.patch_bands, 1)


```

:def yamnet_frames_model(params)

.Defines the YAMNet waveform-to-class-scores model"""

:Args

.params: An instance of Params containing hyperparameters

:Returns

.A model accepting (num_samples,) waveform input and emitting

.predictions: (num_patches, num_classes) matrix of class scores per time frame -

.embeddings: (num_patches, embedding size) matrix of embeddings per time frame -

.log_mel_spectrogram: (num_spectrogram_frames, num_mel_bins) spectrogram feature matrix -

"""

waveform = layers.Input(batch_shape=(None,), dtype=tf.float32)

waveform_padded = features_lib.pad_waveform(waveform, params)

)log_mel_spectrogram, features = features_lib.waveform_to_log_mel_spectrogram_patches

(waveform_padded, params

predictions, embeddings = yamnet(features, params)

```

)frames_model = Model
    ,name='yamnet_frames', inputs=waveform
    (outputs=[predictions, embeddings, log_mel_spectrogram]
return frames_model

def class_names(class_map_csv):
    """Read the class name definition file and return a list of strings"""
    if tf.is_tensor(class_map_csv):
        class_map_csv = class_map_csv.numpy()
    with open(class_map_csv) as csv_file:
        reader = csv.reader(csv_file)
        next(reader) # Skip header
    return np.array([display_name for (_, _, display_name) in reader])

```

: توضیح کد

مرحله اول: ایمپورت ها

```

import csv
import numpy as np
import tensorflow as tf
from tf_keras import Model, layers
import features as features_lib

```

- برای خواندن فایل های CSV مثل class map

- برای عملیات عددی numpy.

- فریم ورک اصلی یادگیری عمیق tensorflow.

- TensorFlow نسخه‌ی موجود در Keras from tf_keras import Model, layers

log-mel-spectrogram کدی که قبلاً بررسی کردیم، مسئول تبدیل waveform به ویژگی‌های features است.

مرحله دوم: توابع داخلی تعریف لایه‌ها

```
def _batch_norm(name, params):  
    def _batch_norm(name, params):  
        def _bn_layer(layer_input):  
            return layers.BatchNormalization(  
                name=name,  
                center=params.batchnorm_center,  
                scale=params.batchnorm_scale,  
                epsilon=params.batchnorm_epsilon)(layer_input)  
        return _bn_layer
```

این تابع یک لایه‌ی BatchNormalization می‌سازد که به نرمال‌سازی خروجی لایه‌ها کمک می‌کند.
مقادیر params.ε، params.scale و params.center را از شیء layers.Conv2D گرفته می‌شوند.

```
def _conv(name, kernel, stride, filters, params):  
    def _conv(name, kernel, stride, filters, params):  
        def _conv_layer(layer_input):  
            output = layers.Conv2D(name='{}/conv'.format(name),  
                filters=filters,  
                kernel_size=kernel,  
                strides=stride,  
                padding=params.conv_padding,  
                use_bias=False,
```

```

activation=None)(layer_input)

output = _batch_norm('{}_conv/bn'.format(name), params)(output)

output = layers.ReLU(name='{}_relu'.format(name))(output)

return output

return _conv_layer

```

- یک لایه‌ی Convolutional ساده تعریف می‌کند.
- شامل Conv2D → BatchNorm → ReLU است.
- این ساختار پایه برای استخراج ویژگی‌ها از تصویر یا اسپکتروگرام است.

def _separable_conv(...):

```

def _separable_conv(name, kernel, stride, filters, params):

def _separable_conv_layer(layer_input):
    output = layers.DepthwiseConv2D(...) # فیلتر جداگانه برای هر کanal#
    ...
    output = layers.Conv2D(...) # ترکیب نتایج کanalها
    ...
    return output

return _separable_conv_layer

```

- این ساختار Separable Convolution است، برای افزایش سرعت و کاهش پارامترها.
- ابتدا DepthwiseConv2D هر کanal را جداگانه پردازش می‌کند.
- سپس Pointwise Conv2D خروجی‌ها را ترکیب می‌کند.
- مشابه MobileNet طراحی شده.

مرحله سوم: ساخت مدل **YAMNet** تابع ()**yamnet()**

```
def yamnet(params):
```

"""\Defines the YAMNet model in Keras."""

```
waveform = layers.Input(batch_shape=(None,), dtype=tf.float32)
```

```
padded_waveform = tf.keras.layers.Lambda(  
    lambda x: features_lib.pad_waveform(x, params),  
    name='waveform_padding')(waveform)
```

```
log_mel_spectrogram, features = tf.keras.layers.Lambda(  
    lambda x: features_lib.waveform_to_log_mel_spectrogram_patches(x, params),  
    name='features')(padded_waveform)
```

```
net = layers.Reshape((features.shape[1], features.shape[2], 1))(features)
```

این بخش چه می کند؟

.1 ورودی مدل : یک موج صوتی (1D Tensor)

.2 .waveform می کند تا wave padding صفر باشد.

.3 .waveform_to_log_mel_spectrogram_patches تبدیل موج صوتی به شبکه ویژگی های قابل استفاده برای (spectrogram)

.4 [batch, frames, mel_bands, 1] به [batch, frames, mel_bands] تبدیل برای ورودی CNN Reshape

مرحله چهارم: پشتیهای لایه های کانولوشن

```
net = _conv('conv1', (3, 3), (2, 1), 32, params)(net)
```

```
net = _separable_conv('conv2', (3, 3), (1, 1), 64, params)(net)
```

```
net = _separable_conv('conv3', (3, 3), (2, 2), 128, params)(net)
```

```
net = _separable_conv('conv4', (3, 3), (1, 1), 128, params)(net)
```

```

net = _separable_conv('conv5', (3, 3), (2, 2), 256, params)(net)
net = _separable_conv('conv6', (3, 3), (1, 1), 256, params)(net)
net = _separable_conv('conv7', (3, 3), (2, 2), 512, params)(net)
net = _separable_conv('conv8', (3, 3), (1, 1), 512, params)(net)

```

این لایه‌ها چه می‌کنند؟

- شبکه‌ای عمیق از `separable conv`, `convolutional layers` برای استخراج ویژگی‌های سطح بالا از `log-mel spectrogram`
- هر لایه می‌توانه ابعاد زمانی یا فرکانسی را کاهش بده و اطلاعات غنی‌تری یاد بگیره.

مرحله پنجم: **Dense** و **Pooling** نهایی

```

net = layers.GlobalAveragePooling2D(name='global_avg_pooling')(net)
net = layers.Dense(units=params.num_classes,
                   activation='softmax',
                   name='prediction')(net)

```

وظیفه‌ی این بخش:

- تمام ویژگی‌ها در ابعاد مکانی را میانگین می‌گیره.
- خروجی با تعداد کلاس‌های مدل `YAMNet`، با `softmax` برای پیش‌بینی دسته‌ی صوتی نهایی.

مرحله ششم: ساخت مدل **Keras**

```
return Model(name='yamnet', inputs=waveform, outputs=net)
```

- این خط در نهایت مدل `Keras` را به نام '`yamnet`' می‌سازد، با ورودی `waveform` و خروجی لیبل‌های صوتی.

تابع (`yamnet_frames_model()`)

```

def yamnet_frames_model(params):
    ...

```

```
scores = yamnet_model(waveform)  
embeddings = embedding_model(waveform)  
...
```

- خروجی مدل رو برای هر فریم زمانی محاسبه می کنه.
- همچنین **embedding** (ویژگی میانی) هم ارائه می دهد برای استفاده در وظایف دیگر.

تابع `class_names(csv_path)`

```
def class_names(csv_path):  
...  
return [entry[2] for entry in csv.reader(csvfile)][1:]
```

- فایل `yamnet_class_map.csv` رو می خونه.
- فقط نام کلاس ها (`label`) رو برمی گردونه.

`params.py:`

```
"""Hyperparameters for YAMNet."""
```

```
from dataclasses import dataclass
```

```
# The following hyperparameters (except patch_hop_seconds) were used to train YAMNet,  
@dataclass(frozen=True) # Instances of this class are immutable.  
class Params:  
    sample_rate: float = 16000.0  
    stft_window_seconds: float = 0.025  
    stft_hop_seconds: float = 0.010  
    mel_bands: int = 64  
    mel_min_hz: float = 125.0
```

```

mel_max_hz: float = 7500.0
log_offset: float = 0.001
patch_window_seconds: float = 0.96
patch_hop_seconds: float = 0.48

@property
def patch_frames(self):
    return int(round(self.patch_window_seconds / self.stft_hop_seconds))

@property
def patch_bands(self):
    return self.mel_bands

num_classes: int = 521
conv_padding: str = 'same'
batchnorm_center: bool = True
batchnorm_scale: bool = False
batchnorm_epsilon: float = 1e-4
classifier_activation: str = 'sigmoid'

tflite_compatible: bool = False

```

: توضیح کد

شامل هایپرپارامترهای مدل YAMNet هست. این پارامترها مستقیماً روی نحوه پردازش سیگنال صوتی و ساختار مدل تأثیر می‌گذارند. توضیحش رو گام به گام و خط به خط شروع می‌کیم:

بخش اول: توضیح اولیه و import

"""Hyperparameters for YAMNet."""

```
from dataclasses import dataclass
```

این فایل برای تعریف های پرپارامترهای اصلی مدل YAMNet استفاده می شود.
از پایتون 3.7 به بعد برای ساخت کلاس های سبک و خوانا به کار می رود.

تعریف کلاس **Params** با دکوراتور **@dataclass**

```
@dataclass(frozen=True) # Instances of this class are immutable.
```

class Params:

@dataclass پایتون به صورت خودکار متدهای `__init__`, `__repr__` و ... رو تولید می کنه.
با عث می شه بعد از ساخت شیء، هیچ کدام از پارامترها قابل تغییر نباشن (ایمن تر).

پارامترهای مربوط به پردازش سیگنال

```
sample_rate: float = 16000.0
```

نرخ نمونه برداری: صدای ورودی با نرخ ۱۶۰۰۰ نمونه در ثانیه ضبط می شه.
این مقدار با مدل اصلی YAMNet سازگار هست

```
stft_window_seconds: float = 0.025
```

طول پنجره: STFT اندازه هر پنجره در تبدیل فوریه کوتاه مدت (STFT)، یعنی ۲۵ میلی ثانیه.

```
stft_hop_seconds: float = 0.010
```

گام بین پنجره ها در STFT یعنی هر ۱۰ میلی ثانیه یک پنجره جدید شروع می شه) حتی اگر قبلی کامل نشده باشه → Overlap وجود دارد

```
mel_bands: int = 64
```

تعداد باندهای Mel: تبدیل به مقیاس فرکانسی Mel انجام می شه، با ۶۴ باند.
مقیاس Mel با ادراک شنوایی انسان هم خوانی دارد

```
mel_min_hz: float = 125.0
```

```
mel_max_hz: float = 7500.0
```

بازه فرکانسی برای Mel:

پایین‌ترین فرکانس: ۱۲۵ هرتز •

بالاترین فرکانس: ۷۵۰۰ هرتز •

`log_offset: float = 0.001`

جلوگیری از لگاریتم صفر: به مقادیر طیفی یک عدد کوچک اضافه می‌شود تا $\log(0)$ پیش نیاد.

پارامترهای مربوط به استخراج ویژگی (patches)

`patch_window_seconds: float = 0.96`

طول هر قطعه ورودی به مدل: مدل با بازه‌های ۰.۹۶ ثانیه‌ای کار می‌کند. یعنی هر ۰.۹۶ ثانیه از صدا، تبدیل به یک ورودی مجزا می‌شود.

`patch_hop_seconds: float = 0.48`

گام حرکت بین قطعات: با هر ۰.۴۸ ثانیه، یک پنجره جدید شروع می‌شود. پس ممکنه دو پنجره هم‌پوشانی داشته باشند.

ویژگی‌های محاسبه شده (Properties)

`@property`

`def patch_frames(self):`

`return int(round(self.patch_window_seconds / self.stft_hop_seconds))`

`patch_frames:`

این متده محاسبه می‌کند که یک قطعه صوتی چند فریم STFT خواهد داشت.

:مثال

با مقادیر پیش‌فرض:

`patch_window_seconds = 0.96`

`stft_hop_seconds = 0.01`

تعداد فریم‌ها در هر patch

`patch_frames = 0.96 / 0.01 = 96`

```

@property
def patch_bands(self):
    return self.mel_bands

```

patch_bands:

تعداد باندهای Mel که در هر فریم وجود دارد (برابر با mel_bands، یعنی ۶۴). این مقدار تعداد ستون‌های ماتریس ویژگی‌هاست.

تنظیمات مربوط به مدل **YAMNet CNN + طبقه‌بندی**

```
num_classes: int = 521
```

تعداد کلاس‌ها: مدل YAMNet قادر است بین ۵۲۱ نوع رویداد صوتی مختلف (بر اساس AudioSet) تمایز قائل شود.

```
conv_padding: str = 'same'
```

نوع **padding** در کانولوشن‌ها:

مقدار 'same' به این معناست که خروجی کانولوشن هم اندازه ورودی باقی می‌ماند با padding خودکار

```
batchnorm_center: bool = True
```

```
batchnorm_scale: bool = False
```

تنظیمات نرمال‌سازی دسته‌ای (**BatchNorm**)

- $\text{center} = \text{True}$: $\text{offset} (\beta)$ عامل

- $\text{scale} = \text{False}$: $\text{scale} (\gamma)$ عامل انجام نمی‌شود (یعنی فقط مرکز کردن).

```
batchnorm_epsilon: float = 1e-4
```

اپسیلون در **BatchNorm**

برای جلوگیری از تقسیم بر صفر در نرمال‌سازی، این مقدار کوچک به واریانس اضافه می‌شود.

```
classifier_activation: str = 'sigmoid'
```

تابع فعال سازی نهایی در مدل طبقه‌بندی:

تابع sigmoid استفاده می‌شود، چون مدل چند-کلاسی و چند-برچسبی (multi-label) است؛ یعنی یک نمونه می‌تواند هم‌زمان به چند کلاس تعلق داشته باشد.

تنظیمات مربوط به TFLite

tflite_compatible: bool = False

سازگاری با TensorFlow Lite

اگر این مقدار True باشد، برخی تنظیمات ساده‌سازی می‌شوند تا مدل قابل تبدیل به TFLite باشد.

جمع‌بندی کلی ساختار کلاس

کلاس Params تمام تنظیمات پردازش سیگنال، تبدیل STFT، تبدیل Mel، ساخت قطعات(patches)، و تنظیمات شبکه عصبی را در خود نگه می‌دارد. این کلاس در کل پروژه YAMNet به عنوان مرجع واحد پارامترها استفاده می‌شود.

.env.php:

```
<?php  
// DB env  
  
define('DB_HOST', 'localhost');  
  
define('DB_NAME', 'iotdash');  
  
define('DB_USER', 'iotapp');  
  
define('DB_PASS', 'ChangeMe_Strong!');  
  
define('DB_CHARSET', 'utf8mb4');
```

: توضیح

1

```
<?php
```

این خط نشان می‌دهد که کد به زبان PHP شروع می‌شود. هر چیزی بعد از این تا زمان بسته شدن <? به عنوان کد PHP اجرا می‌شود.

2

```
// DB env
```

این یک کامنت است (توضیح برای برنامه‌نویس و بی‌اثر در اجرا).
یعنی "Database environment" یا همان محیط متغیرهای پایگاه داده.

3

```
define('DB_HOST', 'localhost');
```

برای تعریف یک ثابت (**constant**) در PHP استفاده می‌شود.

اینجا یک ثابت به اسم DB_HOST ساخته شده.

مقدار آن 'localhost' است.

یعنی پایگاه داده روی همان سیستم (لوکال‌هوست) اجرا می‌شود.

4

```
define('DB_NAME', 'iotdash');
```

یک ثابت دیگر تعریف شده به اسم DB_NAME

مقدار آن 'iotdash' است.

این اسم دیتابیس MySQL است که برنامه باید به آن وصل شود.

5

```
define('DB_USER', 'iotapp');
```

یک ثابت دیگر به اسم DB_USER.

مقدار آن 'iotapp' است.

این نام کاربری (username) پایگاه داده است که برای اتصال استفاده می‌شود.

6

```
define('DB_PASS', 'ChangeMe_Strong!');
```

یک ثابت به اسم DB_PASS.

مقدار آن 'ChangeMe_Strong!'.

این رمز عبور (password) کاربر پایگاه داده است.

7

```
define('DB_CHARSET', 'utf8mb4');
```

یک ثابت به اسم DB_CHARSET.

مقدار آن 'utf8mb4'.

این یعنی وقتی به دیتابیس وصل می‌شویم، از کاراکترست UTF-8 با پشتیبانی کامل (حتی ایموجی‌ها) استفاده شود.

خلاصه: این کد فقط یک سری ثابت‌های محیطی (Environment Constants) برای اتصال به دیتابیس MySQL تعریف می‌کند:

میزبان دیتابیس localhost

نام دیتابیس iotdash

یوزر دیتابیس iotapp

پسورد یوزر ChangeMe_Strong!

کدگذاری کاراکترها utf8mb4

_top.php:

```
<?php
session_start(); require_once __DIR__.'/config.php';

if (empty($_SESSION['uid'])) { header('Location: index.php'); exit; }

$prev = $_SESSION['last_login_prev'] ?? null;
$user = $_SESSION['username'] ?? 'کاربر';

?>

<nav class="navbar navbar-expand-lg navbar-blur shadow-sm sticky-top">
<div class="container-fluid">
<!!-- نماد خوشگل -->
<a class="navbar-brand fw-bold d-flex align-items-center"
href="dashboard.php">
<i class="bi bi-lightning-charge-fill text-warning me-2 fs-4"></i>
IoT Dash
</a>
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item"><a class="nav-link" href="dashboard.php"><i class="bi bi-speedometer2 me-1"></i>داشبورد</a></li>
<li class="nav-item"><a class="nav-link" href="events.php"><i class="bi bi-mic me-1"></i>رویدادهای صوتی</a></li>
<li class="nav-item"><a class="nav-link" href="processes.php"><i class="bi bi-activity me-1"></i>پردازهها</a></li>
<li class="nav-item"><a class="nav-link" href="metrics.php"><i class="bi bi-graph-up-arrow me-1"></i>متريکها</a></li>
```

```

<li class="nav-item"><a class="nav-link" href="system.php"><i class="bi bi-pc-display me-1"></i>مشخصات سیستم</a></li>

</ul>

<div class="d-flex align-items-center gap-3">

<span class="text-light small">

<?= htmlspecialchars($user) ?> | خوش آمدید
<?= $prev ? htmlspecialchars($prev) : 'نخستین بار' ?> خرین ورود

</span>

<span id="clock" class="badge bg-light text-dark"></span>

<a class="btn btn-sm btn-outline-danger" href="logout.php"><i class="bi bi-box-arrow-right"></i></a>

</div>

</div>

</nav>

<script>

setInterval(()=>{ document.getElementById('clock').textContent = new Date().toLocaleString('fa-IR'); },1000);

</script>

```

توضیح :

بخش PHP بالای فایل

```

<?php
session_start(); require_once __DIR__.'/config.php';

```

- سیشن (Session) جدید باز می‌کند یا سشن موجود را ادامه می‌دهد. این برای ذخیره اطلاعات کاربر (مثل ورود به سیستم) در تمام صفحات استفاده می‌شود.
 - فایل config.php که تنظیمات (مثل اتصال به دیتابیس) دارد را require_once __DIR__.'/config.php'; بارگذاری می‌کند __DIR__. مسیر جاری فایل را برمی‌گرداند تا همیشه آدرس درست باشد.
- ```
if (empty($_SESSION['uid'])) { header('Location: index.php'); exit; }
```
- چک می‌کند اگر کلید uid در سشن خالی باشد (یعنی کاربر لایگین نکرده)، او را با دستور header() به صفحه‌ی index.php (صفحه‌ی ورود) بفرستد و با exit; اجرای کد متوقف شود.
- ```
$prev = $_SESSION['last_login_prev'] ?? null;
```
- ```
$user = $_SESSION['username'] ?? 'کاربر' ;
```
- مقدار آخرین زمان ورود (last\_login\_prev) را از سشن می‌گیرد. اگر وجود نداشته باشد مقدار null قرار می‌دهد.
  - نام کاربر (username) را می‌گیرد و اگر وجود نداشته باشد پیش‌فرض "کاربر" می‌گذارد.

## بخش HTML ناوابار

```
<nav class="navbar navbar-expand-lg navbar-blur shadow-sm sticky-top">
```

```
<div class="container-fluid">
```

- یک navbar (نوار منو بالای سایت) با کلاس‌های Bootstrap ساخته می‌شود.
- باعث می‌شود navbar همیشه در بالای صفحه بماند.
- container-fluid یعنی عرضش کل صفحه را پر کند.

```

```

```
<i class="bi bi-lightning-charge-fill text-warning me-2 fs-4"></i>
```

IoT Dash

</a>

- لینک لوگوی سایت (IoT Dash) است.

آیکون رعد (bi-lightning-charge-fill) از Bootstrap Icons اضافه شده.

وقتی کلیک شود به صفحه dashboard.php می‌رود.

```
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
 <li class="nav-item"><i class="bi bi-speedometer2 me-1"></i>داشبورد
 <li class="nav-item"><i class="bi bi-mic me-1"></i>رویدادهای صوتی
 <li class="nav-item"><i class="bi bi-activity me-1"></i>پردازهها
 <li class="nav-item"><i class="bi bi-graph-up-arrow me-1"></i>متريکها
 <li class="nav-item"><i class="bi bi-pc-display me-1"></i>مشخصات سیستم

```

- لیست منوی سایت:

◦ داشبورد → صفحه اصلی آمار.

◦ رویدادهای صوتی → صفحه مربوط به لاغ صداها.

◦ پردازهها → نمایش پردازههای سیستم.

◦ متريکها → نمودارها و دادههای مصرف سیستم.

◦ مشخصات سیستم → مشخصات دستگاه OS، CPU، RAM.

• هر لینک با یک آیکون مناسب نمایش داده می‌شود.

## بخش سمت راست نوار

```
<div class="d-flex align-items-center gap-3">

 <?= htmlspecialchars($user) ?> | خوش آمدید
 <?= $prev ? htmlspecialchars($prev) : 'آخرین ورود' ?>

```

- یک متن کوچک در سمت راست نمایش داده می‌شود:
  - اسم کاربر (\$user) با htmlspecialchars مان چاپ می‌شود.
- بعد از آن پیام خوشآمد و زمان آخرین ورود (\$prev) نمایش داده می‌شود. اگر خالی باشد، عبارت «آخرین بار» نشان داده می‌شود.

```

```

- یک المان برای نمایش ساعت زنده در navbar.
- استایلش مثل یک badge (کادر کوچک) با زمینه روشن است.

```
<i class="bi bi-box-arrow-right" style="font-size: 1.5em;"></i>
```

- یک دکمه قرمز کوچک برای خروج (logout.php).
- آیکون bi-box-arrow-right نشان‌دهنده خروج است.

## اسکریپت جاوا اسکریپت

```
<script>
 setInterval(()=>{ document.getElementById('clock').textContent = new
 Date().toLocaleString('fa-IR'); },1000);
</script>
```

- هر 1 ثانیه (1000ms) یک بار ساعت سیستم به زبان فارسی (fa-IR) گرفته و در عنصر `#clock` نمایش داده می‌شود.
- این باعث می‌شود ساعت زنده همیشه در بالای سایت باشد.

## خلاصه

این کد یک **navbar** داینامیک و اینمن برای داشبورد IoT شما می‌سازد که:

- وضعیت ورود کاربر را بررسی می‌کند.
- نام کاربر و آخرین ورود را نمایش می‌دهد.
- منوی کامل به صفحات مختلف (داشبورد، رویدادها، پردازه‌ها، متريک‌ها، مشخصات) دارد.
- ساعت زنده را نشان می‌دهد.
- خروج امن دارد.

### `audio_blocklist.php:`

```
<?php
// audio_blocklist.php — نسخه هماهنگ با هدر سایت —
(اگر لاغ خطای خواستی، این دو خط را موقتاً باز کن) //
// error_reporting(E_ALL);
// ini_set('display_errors', 1);
```

نوار بالا + اسکلت صفحه `include '_top.php'; //`

`// مسیرها`

```
$CSV_PATH = __DIR__ . '/assets/yamnet_class_map.csv';
$CFG_DIR = __DIR__ . '/config';
$BLOCK_PATH = $CFG_DIR . '/blocklist.json';
```

```

// lowercase امن بدون mbstring

function lower($s){ return function_exists('mb_strtolower') ? mb_strtolower($s,'UTF-8') : strtolower($s); }

// اطمینان از وجود پوشه config
if (!is_dir($CFG_DIR)) { @mkdir($CFG_DIR, 0755, true); }

// خواندن بلاکلیست
$blocked = [];

if (is_file($BLOCK_PATH)) {
 $j = @file_get_contents($BLOCK_PATH);
 $tmp = @json_decode($j, true);
 if (is_array($tmp)) $blocked = $tmp;
}

// ذخیره فرم
$msg = "";

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['labels']) && is_array($_POST['labels'])) {
 $new = array_values(array_unique(array_map('trim', $_POST['labels'])));
 @file_put_contents($BLOCK_PATH, json_encode($new,
 JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT));
 $blocked = $new;
 $msg = 'تغییرات با موفقیت ذخیره شد';
}

}

```

```

// آپلود CSV

if (isset($_POST['upload_csv']) && isset($_FILES['csv']) && $_FILES['csv']['error'] ===
UPLOAD_ERR_OK) {

 @move_uploaded_file($_FILES['csv']['tmp_name'], $CSV_PATH);

 $msg = 'با موفقیت بارگذاری شد CSV فایل';

}

// خواندن CSV

$labels = [];

if (is_file($CSV_PATH)) {

 if (($h = @fopen($CSV_PATH, 'r')) !== false) {

 $header = @fgetcsv($h);

 $idx = null;

 if ($header && is_array($header)) {

 foreach ($header as $i => $col) {

 $c = lower(trim((string)$col));

 if (in_array($c, ['display_name','name','label'], true)) { $idx = $i; break; }

 }

 }

 if ($idx === null) {

 if ($header && count($header) === 1) { $labels[] = trim((string)$header[0]); }

 elseif ($header) { $labels[] = trim(implode(',', $header)); }

 while (($row = @fgetcsv($h)) !== false) {

 if (!$row) continue;

 $labels[] = trim(count($row) === 1 ? (string)$row[0] : implode(',', $row));

 }

 }

 }

}


```

```

 }

} else {

 while (($row = @fgetcsv($h)) !== false) {

 if (!isset($row[$idx])) continue;

 $name = trim((string)$row[$idx]);

 if ($name !== "") $labels[] = $name;

 }

}

@fclose($h);

}

$labels = array_values(array_filter(array_unique($labels), fn($x)=>$x!=""));

natcasesort($labels);

$labels = array_values($labels);

}

?>

```

<!-- لینک‌های استایل مثل سایر صفحات -->

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css?v=2">

<style>

/* فقط ریز تنظیم‌های همین صفحه */

.card-desc{color:#9aa4b2;font-size:.9rem}

```

```

.sticky-
actions{position:sticky;bottom:0;background:transparent;padding:.75rem;border-top:1px
solid rgba(255,255,255,.08)}

.search-input{max-width:420px}

.label-row{display:flex;align-items:center;gap:.5rem;padding:.35rem
.5rem;border-bottom:1px dashed rgba(255,255,255,.12)}

.label-row:hover{background:rgba(255,255,255,.03)}

</style>

```

```

<div class="container py-4">

<div class="d-flex align-items-center gap-2 mb-3">

<i class="bi bi-x-octagon fs-4"></i>

<h5 class="mb-0 fw-bold">محدودسازی رویدادها در تشخیص صدا</h5>

</div>

<p class="card-desc mb-3">

موارد تیک خورده در تشخیص صوت نادیده گرفته می شوند.

CSV فایل <code>assets/yamnet_class_map.csv</code> خوانده می شود.

</p>

<?php if (!empty($msg)): ?>

<div class="alert alert-success py-2"><?php echo htmlspecialchars($msg);
?></div>

<?php endif; ?>

<?php if (!is_file($CSV_PATH)): ?>

<div class="alert alert-warning">CSV فایل پیدا نشد. لطفاً آپلود کنید.</div>

<form method="post" enctype="multipart/form-data" class="mb-4">
```

```

<input type="hidden" name="upload_csv" value="1">

<div class="input-group" style="max-width:520px">
 <input type="file" name="csv" class="form-control" accept=".csv" required>
 <button class="btn btn-primary">آپلود CSV</button>
</div>
</form>
<?php endif; ?>

<?php if ($labels): ?>
 <div class="d-flex justify-content-between align-items-center mb-2">
 <input id="q" class="form-control form-control-sm search-input"
placeholder="...جستجوی نام رویداد">
 <div class="d-flex gap-2">
 <button type="button" class="btn btn-sm btn-outline-secondary"
id="btnAll">انتخاب همه</button>
 <button type="button" class="btn btn-sm btn-outline-secondary"
id="btnNone">عدم انتخاب</button>
 <button type="button" class="btn btn-sm btn-outline-secondary"
id="btnInvert">معکوس</button>
 </div>
 </div>
</div>

<form method="post" id="frm">
 <div class="card">
 <div class="card-body p-0" id="list" style="max-height:60vh;overflow:auto;">
 <?php foreach ($labels as $name):>

```

```

$checked = in_array($name, $blocked, true);

?>

<label class="label-row">

 <input type="checkbox" class="form-check-input me-1" name="labels[]" value="<?php echo htmlspecialchars($name); ?>" <?php echo $checked?'checked':''; ?>>

 <?php echo htmlspecialchars($name); ?>

</label>

<?php endforeach; ?>

</div>

<div class="sticky-actions d-flex justify-content-between align-items-center">

 <small class="text-muted">تعداد کل:<?php echo count($labels); ?> · انتخاب شده:<?php echo $selCnt; ?></small>

 <button class="btn btn-danger">ذخیره تغییرات</button>

</div>

</div>

</form>

<?php endif; ?>

<div class="mt-3">

 <i class="bi bi-arrow-right-circle"></i> بازگشت به داشبورد

</div>

</div>

<script src="assets/app.js"></script>

<script>

```

// منتقلش کنی app.js می تونی به) اسکریپت همین صفحه

```

(function(){

 const q=document.getElementById('q'),list=document.getElementById('list'),
 btnAll=document.getElementById('btnAll'),btnNone=document.getElementById('btnNone'),
 btnInvert=document.getElementById('btnInvert'),selCnt=document.getElementById('selCnt');

 function updateCount(){ if(!list) return;
 selCnt.textContent=list.querySelectorAll('input[type="checkbox"]:checked').length; }

 updateCount();
 list?.addEventListener('change',updateCount);

 q?.addEventListener('input',()=>{ const t=(q.value||"").trim().toLowerCase();
 list.querySelectorAll('.label-row').forEach(row=>{
 const name=row.querySelector('span').textContent.toLowerCase();
 row.style.display=name.includes(t)?'none';
 });
 });

 btnAll?.addEventListener('click',()=>{
 list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=true);
 updateCount(); });

 btnNone?.addEventListener('click',()=>{
 list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=false);
 updateCount(); });

 btnInvert?.addEventListener('click',()=>{
 list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=!cb.checked);
 updateCount(); });

})();
</script>
```

توضیح :

## بخش PHP بالای فایل

### اهدار و عیب‌یابی اختیاری

```
// audio_blocklist.php — نسخه هماهنگ با هدر سایت
// error_reporting(E_ALL);
// ini_set('display_errors', 1);
```

- توضیح/برچسب فایل.

- دو خط کامنت شده: اگر موقع توسعه خواستی خطاه را ببینی، همین دو خط را موقتاً از کامنت دربیار تا همه خطاهای نمایش داده شود (در محیط تولید بهتره خاموش باشه).

### ۲) اضافه کردن اسکلت صفحه

```
include '_top.php'; // نوار بالا + اسکلت صفحه
```

- فایل مشترک هدر/ناویار/شروع <html> را می‌آورد تا ظاهر صفحه با بقیه‌ی سایت یکی باشد و کنترل سشن/احراز هویت همان‌جا انجام شود.

### ۳) تعریف مسیرها

```
$CSV_PATH = __DIR__ . '/assets/yamnet_class_map.csv';
$CFG_DIR = __DIR__ . '/config';
$BLOCK_PATH = $CFG_DIR . '/blocklist.json';
```

- مسیر CSV کلاس‌های YAMNet، پوشه‌ی تنظیمات(config)، و فایل ذخیره‌ی بلاکلیست(blocklist.json) را تعیین می‌کند \_\_DIR\_\_. یعنی پوشه‌ی همین فایل.

### ۴) تابع lowercase امن

```
function lower($s){ return function_exists('mb_strtolower') ? mb_strtolower($s,'UTF-8') : strtolower($s); }
```

- یک نرمال‌ساز حروف کوچک: اگر mb\_strtolower یونی‌کد UTF-8 موجود بود از آن استفاده می‌کند؛ وگرنه از strToLower معمولی. برای اینکه ستون‌های CSV با «Display\_Name / Name / Label» به شکل «strToLower» حساسیت به بزرگی حروف مقایسه شوند.

## ۵ (اطمینان از وجود پوشه‌ی config)

```
if (!is_dir($CFG_DIR)) { @mkdir($CFG_DIR, 0755, true); }
```

- اگر پوشه‌ی config وجود نداشت، آن را با دسترسی 0755 می‌سازد @. خطای احتمالی را بی‌صدا می‌کند.

## ۶ (خواندن بلاک‌لیست موجود)

```
$blocked = [];

if (is_file($BLOCK_PATH)) {

 $j = @file_get_contents($BLOCK_PATH);

 $tmp = @json_decode($j, true);

 if (is_array($tmp)) $blocked = $tmp;

}

$blocked
```

- اگر blocklist.json قبلاً وجود داشته باشد، خوانده می‌شود و اگر JSON معتبر آرایه بود، در \$blocked می‌نشینند. این آرایه اسم کلاس‌هایی است که کاربر قبلًاً تیک زده و باید نادیده گرفته شوند.

## ۷ ذخیره فرم (ارسال بلاک‌لیست جدید)

```
$msg = ";

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['labels']) &&
is_array($_POST['labels'])) {

 $new = array_values(array_unique(array_map('trim', $_POST['labels'])));

 @file_put_contents($BLOCK_PATH, json_encode($new,
JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT));

 $blocked = $new;

 $msg = '!تغییرات با موفقیت ذخیره شد';

}
```

- اگر فرم با متد POST ارسال شده و labels[] آرایه‌ای باشد:

- ورودی‌ها trim و unique می‌شوند (حذف فضاهای اضافی و موارد تکراری).

- در JSON یونیکد و مرتب ذخیره می‌شوند.

• \$blocked روز می‌شود.

◦ پیام موفقیت برای نمایش آماده می‌شود.

#### ۸ آپلود CSV جدید

```
if (isset($_POST['upload_csv']) && isset($_FILES['csv']) && $_FILES['csv']['error'] ===
UPLOAD_ERR_OK) {

 @move_uploaded_file($_FILES['csv']['tmp_name'], $CSV_PATH);

 $msg = 'با موفقیت بارگذاری شد CSV فایل';

}

@move_uploaded_file($_FILES['csv']['tmp_name'], $CSV_PATH);
$msg = 'با موفقیت بارگذاری شد CSV فایل';
}

@move_uploaded_file($_FILES['csv']['tmp_name'], $CSV_PATH);
$msg = 'با موفقیت بارگذاری شد CSV فایل';
}
```

• اگر دکمه‌ی آپلود CSV زده شده و فایل بدون خطا آپلود شده:

◦ فایل موقت به مسیر نهایی assets/yamnet\_class\_map.csv منتقل می‌شود.

◦ پیام موفقیت برای نمایش تنظیم می‌شود.

#### ۹ خواندن CSV و استخراج لیبل‌ها

```
$labels = [];

if (is_file($CSV_PATH)) {

 if (($h = fopen($CSV_PATH, 'r')) !== false) {

 $header = fgetcsv($h);

 $idx = null;

 if ($header && is_array($header)) {

 foreach ($header as $i => $col) {

 $c = strtolower(trim((string)$col));

 if (in_array($c, ['display_name', 'name', 'label'], true)) { $idx = $i; break; }

 }

 }

 }

}
```

• اگر CSV موجود باشد، بازش می‌کند.

- ردیف اول (header) را می‌خواند و دنبال ستونی با نام «name» یا «display\_name» یا «label» (بدون حساسیت به حروف) می‌گردد. اگر یافت شود، شماره‌ی ستون در \$idx ذخیره می‌شود.

```

if ($idx === null) {

 if ($header && count($header) === 1) { $labels[] = trim((string)$header[0]); }

 elseif ($header) { $labels[] = trim(implode(',', $header)); }

 while (($row = @fgetcsv($h)) !== false) {

 if (!$row) continue;

 $labels[] = trim(count($row) === 1 ? (string)$row[0] : implode(',', $row));

 }

} else {

 while (($row = @fgetcsv($h)) !== false) {

 if (!isset($row[$idx])) continue;

 $name = trim((string)$row[$idx]);

 if ($name !== "") $labels[] = $name;

 }

}

@fclose($h);

}

$labels = array_values(array_filter(array_unique($labels), fn($x)=>$x!=""));

natcasesort($labels);

$labels = array_values($labels);

}

```

- اگر ستون هدف پیدا نشد:
- فرض می‌گیرد CSV تکستونه است؛ در غیر این صورت کل ردیف را با کاما به یک رشته تبدیل می‌کند (تا چیزی از دست نرود).  
Fallback

- اگر ستون هدف پیدا شد:

- فقط همان ستون را از هر ردیف می‌خواند و اگر خالی نبود به لیست اضافه می‌کند.
- در پایان:
- حذف خالی‌ها filter و unique
- برای مرتب‌سازی طبیعی بدون حساسیت به حروف، natcasesort
- برای مرتب‌سازی ایندکس‌ها ایندکس‌reindex (array\_values) و.

### بخش استایل‌ها CSS محلی همین صفحه

```
<link ... bootstrap ...>
<link ... bootstrap-icons ...>
<link rel="stylesheet" href="assets/style.css?v=2">
• .. آیکون‌ها، و استایل سراسری سایت Bootstrap بارگذاری
.card-desc{color:#9aa4b2;font-size:.9rem}
.sticky-
actions{position:sticky;bottom:0;background:transparent;padding:.75rem;border-top:1px
solid rgba(255,255,255,.08)}
.search-input{max-width:420px}
.label-row{display:flex;align-items:center;gap:.5rem;padding:.35rem .5rem;border-
bottom:1px dashed rgba(255,255,255,.12)}
.label-row:hover{background:rgba(255,255,255,.03)}
```

### ● ریزتنظیم‌های همین صفحه:

- توضیحات کارت، ناحیه‌ی دکمه‌های پایین کارت با position: sticky همیشه در دید.
- حداکثر عرض جست‌وجو.
- ظاهر هر ردیف لیبل (چک‌باکس + نام).

### HTML محتوا

## عنوان و توضیح

- هدر آیکون و تیتر: محدودسازی رویدادها در تشخیص صدا
- پاراگراف توضیح: تیک خوردها نادیده گرفته می‌شوند؛ CSV از مسیر assets/yamnet\_class\_map.csv.

## ۲ نمایش پیام موفقیت (در صورت وجود)

```
<?php if (!empty($msg)): ?>
 <div class="alert alert-success py-2"><?php echo htmlspecialchars($msg);
?></div>
<?php endif; ?>
```

- اگر \$msg مقدار داشته باشد، به صورت alert موفقیت نشان داده می‌شود.

## ۳ درخواست آپلود CSV اگر فایل موجود نیست

```
<?php if (!is_file($CSV_PATH)): ?>
 <div class="alert alert-warning">فایل CSV کنید آپلود نشد. لطفاً.</div>
 <form method="post" enctype="multipart/form-data" class="mb-4">
 <input type="hidden" name="upload_csv" value="1">
 <div class="input-group" style="max-width:520px">
 <input type="file" name="csv" class="form-control" accept=".csv" required>
 <button class="btn btn-primary">آپلود CSV</button>
 </div>
 </form>
<?php endif; ?>
```

- اگر CSV فعلاً وجود ندارد، پیام هشدار و فرم آپلود نشان داده می‌شود.
- برای آپلود فایل ضروری است. enctype="multipart/form-data"

## ۴ لیست لیبل‌ها + فرم بلاک‌لیست

```
<?php if ($labels): ?>
```

```

<div class="d-flex ...">
 <input id="q" ... placeholder="...جستجوی نام رویداد">
 <div class="d-flex gap-2">
 <button id="btnAll">انتخاب همه</button>
 <button id="btnNone">عدم انتخاب</button>
 <button id="btnInvert">معکوس</button>
 </div>
</div>

<form method="post" id="frm">
 <div class="card">
 <div class="card-body p-0" id="list" style="max-height:60vh;overflow:auto;">
 <?php foreach ($labels as $name):>
 $checked = in_array($name, $blocked, true);
 ?>
 <label class="label-row">
 <input type="checkbox" class="form-check-input me-1" name="labels[]" value="<?php echo htmlspecialchars($name); ?>" <?php echo $checked?'checked':''; ?>>
 <?php echo htmlspecialchars($name); ?>
 </label>
 <?php endforeach; ?>
 </div>
 <div class="sticky-actions d-flex justify-content-between align-items-center">
 <small class="text-muted">تعداد کل: <?php echo count($labels); ?><small> انتخاب شده . 0</small>
 <button class="btn btn-danger">ذخیره تغییرات</button>
 </div>
</form>

```

```
</div>
</div>
</form>
<?php endif; ?>
```

اگر آرایه‌ی \$labels خالی نباشد:

- یک input جستجو برای فیلتر کردن لیبل‌ها.
- سه دکمه‌ی کمکی: «انتخاب همه»، «عدم انتخاب»، «معکوس».
- فرم POST که مجموعه چک‌باکس‌ها را با نام [labels] ارسال می‌کند.
- برای هر لیبل یک ردیف ساخته می‌شود و اگر قبلًا در \$blocked بود، checked می‌شود.
- پایین کارت یک نوار چسبان دارد که شمارنده انتخاب‌شده‌ها و دکمه‌ی «ذخیره تغییرات» را نشان می‌دهد.
- لینک بازگشت به داشبورد هم در انتهای قرار دارد.

#### بخش جاوا اسکریپت (پایین صفحه)

#### ابارگذاری اسکریپت اصلی سایت

```
<script src="assets/app.js"></script>
```

اگر توابع استایل‌های مشترک داشته باشی اینجا load می‌شوند.

#### ۱۲۲ سکریپت مخصوص همین صفحه

```
(function(){
 const q=document.getElementById('q'),list=document.getElementById('list'),

 btnAll=document.getElementById('btnAll'),btnNone=document.getElementById('btnNone'),
,

 btnInvert=document.getElementById('btnInvert'),selCnt=document.getElementById('selCn
t');
```

```

function updateCount(){
 if(!list) return;

 selCnt.textContent=list.querySelectorAll('input[type="checkbox"]:checked').length;
}

updateCount();

list?.addEventListener('change',updateCount);

q?.addEventListener('input',()=>{
 const t=(q.value||"").trim().toLowerCase();
 list.querySelectorAll('.label-row').forEach(row=>{
 const name=row.querySelector('span').textContent.toLowerCase();
 row.style.display=name.includes(t)?'none';
 });
});

btnAll?.addEventListener('click',()=>{
 list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=true);
 updateCount();
});

btnNone?.addEventListener('click',()=>{
 list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=false);
 updateCount();
});

btnInvert?.addEventListener('click',()=>{

```

```

list.querySelectorAll('input[type="checkbox"]').forEach(cb=>cb.checked=!cb.checked);

updateCount();

});

})();

```

- متغیرها را از DOM می‌گیرد.
- تعداد چکباکس‌های تیک‌خورده را می‌شمارد و داخل `#selCnt` می‌گذارد (اولین بار هم یک بار updateCount() اجرا می‌شود).
- روی هر تغییر در لیست چکباکس‌ها، شمارنده به روز می‌شود.
- با تایپ در جست‌وجو (#q)، هر ردیف که نامش شامل متن فیلتر نیست مخفی می‌شود.
- سه دکمه‌ی کنترلی: انتخاب همه، عدم انتخاب، و برعکس کردن انتخاب‌ها—و بعد از هر کدام شمارنده به روز می‌شود.

#### نکات ایمنی و بهبودها (پیشنهادی)

- **CSRF Token**: برای فرم‌ها یک توکن CSRF بگذار تا ارسال جعلی فرم از بیرون ممکن نباشد.
- **CSV**: محدودیت نوع فایل همین حالا accept=".csv" هست، ولی بهتره سرور هم MIME پسوند را چک کند و اندازه فایل را محدود کند.
- دسترسی فایل‌ها: بررسی کن `/config` و `blocklist.json` از وب قابل دانلود نباشند با `.htaccess`. یا نگهداری خارج از وب‌رشته
- **Handy UX**: یک «نشان‌گر انتخاب‌شده‌ها» به صورت live هست؛ اگر خواستی می‌توانی «نشان‌گر فیلترشده‌ها» هم اضافه کنی (مثلاً تعداد آیتم‌های در حال نمایش).
- **ولیدیشن داده‌ها**: قبل از ذخیره، لیبل‌ها را به رشته‌های کوتاه با `whitelist` کارکترها نرمال کنید تا ورودی ناخواسته ذخیره نشود.

#### خروجی سمت مدل تشخیص

- برنامه‌ی تشخیص صدا (Python/YAMNet) می‌توانه قبل از گزارش دادن رویداد، بررسی کنه اگر label در blocklist.json بود، آن رویداد را به کل نادیده بگیرد در DB ننویسد و MQTT منتشر نکند این فایل همین صفحه می‌سازد/به روز می‌کند.

### audio\_settings.php:

```
<?php

require_once __DIR__.'/top.php'; // فرض: این فایل سشن/احراز هویت و هدر را مدیریت می‌کند

require_once __DIR__.'/config.php';

$defaults = [
 'use_ml' => TRUE, // ML حالت پیشفرض: بدون
 'sample_sec' => 12 // پیشفرض: 10 ثانیه
];

function load_audio_cfg($defaults){
 if (defined('AUDIO_CFG_FILE') && file_exists(AUDIO_CFG_FILE)) {
 $j = json_decode(@file_get_contents(AUDIO_CFG_FILE), true);
 if (is_array($j)) return array_replace($defaults, $j);
 }
 return $defaults;
}

function save_audio_cfg($arr){
```

```

if (!defined('AUDIO_CFG_FILE')) return false;

 @file_put_contents(AUDIO_CFG_FILE, json_encode($arr,
JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT));

 return true;
}

$cfg = load_audio_cfg($defaults);

$ok = "";
$err = "";

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

 // امن سازی ورودی ها

 $use_ml = isset($_POST['use_ml']) && $_POST['use_ml'] === '1';

 $allowed = range(5, 20); // 5..20

 $sample = intval($_POST['sample_sec'] ?? $cfg['sample_sec']);

 if (!in_array($sample, $allowed, true)) {

 $err = 'بازه‌ی نمونه برداری نامعتبر است.';

 } else {

 $cfg['use_ml'] = $use_ml;

 $cfg['sample_sec'] = $sample;

 if (save_audio_cfg($cfg)) {

 $ok = 'تنظیمات با موفقیت ذخیره شد.';

 } else {

 $ok = 'تنظیمات با موفقیت ذخیره شد.';

 }

 }

}


```

```

}

}

?>

<!doctype html>

<html lang="fa" dir="rtl">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>تنظیمات تشخیص صدا</title>

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css?v=2">

</head>

<body class="bg-pattern">

<div class="container py-4">

<div class="row justify-content-center">

<div class="col-lg-6">

<div class="card shadow-sm">

<div class="card-header">

<i class="bi bi-sliders2 me-1"></i> تنظیمات تشخیص صدا

</div>

<div class="card-body">

<?php if($ok): ?>

```

```

<div class="alert alert-success py-2"><?=htmlspecialchars($ok)?></div>

<?php endif; ?>

<?php if($err): ?>

<div class="alert alert-danger py-2"><?=htmlspecialchars($err)?></div>

<?php endif; ?>

<form method="post" class="vstack gap-3">

<!-- يا ساده ML حالت -->

<div>

<label class="form-label fw-bold">حالت تشخیص</label>

<div class="form-check form-switch">

<input class="form-check-input" type="checkbox" id="use_ml"
name="use_ml" value="1"

<?= $cfg['use_ml'] ? 'checked' : '' ?>>

<label class="form-check-label" for="use_ml">

(برای دقت بالاتر توصیه می‌شود، اما منابع بیشتری مصرف می‌کند) ML استفاده از مدل

</label>

</div>

<div class="form-text">

سبک‌تر و سریع‌تر است ML استفاده کنید که نیاز به دقت بیشتر دارید؛ حالت بدون ML زمانی از

</div>

</div>

<!-- بازه نمونه برداری 5..20 ثانیه (انتخاب تکی) -->

<div>

```

```

<label class="form-label fw-bold">زمان نمونهبرداری (ثانیه)</label>

<div class="d-flex flex-wrap gap-2">
 <?php foreach (range(5,20) as $sec): ?>
 <div class="form-check">
 <input class="form-check-input" type="radio" name="sample_sec"
 id="s<?=$sec?>" value="<?=$sec?>" <?= ($cfg['sample_sec'] == $sec) ?
 'checked' : " ?>>
 <label class="form-check-label" for="s<?=$sec?>"><?=$sec?></label>
 </div>
 <?php endforeach; ?>
</div>

<div class="form-text">این مقدار، طول «گوشدادن» در هر بار نمونهبرداری را تعیین می‌کند.</div>
</div>

<div class="d-flex gap-2">
 <button class="btn btn-primary"><i class="bi bi-save me-1"></i> ذخیره
 <?php echo $cfg['sample_sec']; ?>
 </button>
 بازگشت به داشبورد
</div>
</form>

</div>
</div>

-- نمایش وضعیت فعلی برای اطمینان -->

```

```

<div class="card mt-3 shadow-sm">
 <div class="card-body">
 <div class="small text-muted">وضعیت فعلی</div>
 <ul class="mb-0">
 <?= $cfg['use_ml'] ? 'ML' : 'ساده' ?>
 <?= intval($cfg['sample_sec']) ?> ثانیه
 <li class="text-muted">فایل <code><?= htmlspecialchars(AUDIO_CFG_FILE) ?></code>

 </div>
</div>

</div>
</div>
</div>
</body>
</html>

```

توضیح :

## بخش PHP منطق بک‌اند

1

```

require_once __DIR__.'/top.php';
require_once __DIR__.'/config.php';

```

• require\_once فایل‌های خارجی را بارگذاری می‌کند.

- مسیر فolder فعلی فایل است.
- 'احتمالاً مدیریت سشن و لگین و هدر صفحه را انجام می‌دهد.' \_top.php'
- تنظیمات مثل نام دیتابیس، یوزر و رمز را نگه می‌دارد. config.php

**2**

```
$defaults = [
 'use_ml' => TRUE,
 'sample_sec' => 12
];
```

- یک آرایه پیش‌فرض ساخته می‌شود.
- مشخص می‌کند آیا از مدل **یادگیری ماشین (ML)** استفاده شود یا خیر.
- تعداد ثانیه‌های پیش‌فرض برای نمونه‌برداری است (ابنجا 12).

**3**

```
function load_audio_cfg($defaults){ ... }
```

تابعی برای خواندن تنظیمات ذخیره‌شده:

- اگر ثابت **AUDIO\_CFG\_FILE** تعریف شده باشد و فایل وجود داشته باشد:
- محتوا خوانده می‌شود (**file\_get\_contents**) و با **json\_decode** آرایه تبدیل می‌شود.
- در صورت معتبر بودن، روی پیش‌فرض‌ها اعمال می‌شود.
- در غیر این صورت همان پیش‌فرض‌ها برگردانده می‌شود.

**4**

```
function save_audio_cfg($arr){ ... }
```

تابعی برای ذخیره تنظیمات در فایل: **JSON**

- اگر ثابت AUDIO\_CFG\_FILE تعریف نشده باشد، برمی‌گرداند.
- اگر تعریف شده باشد:
  - JSON تبدیل می‌کند (با فرمت زیبا و پشتیبانی از یونیکد).
  - در فایل می‌نویسد. (file\_put\_contents).

## 5

```
$cfg = load_audio_cfg($defaults);
$ok = '';
$err = '';
```

- تنظیمات بارگذاری می‌شوند.
- دو متغیر برای پیام موفقیت (\$ok) و خطا (\$err) آماده می‌شوند.

## 6

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 ...
}
```

این بلوک فقط وقتی اجرا می‌شود که فرم توسط کاربر ارسال (**Submit**) شود.

- ورودی‌ها امن‌سازی می‌شوند.
- checkbox با use\_ml تعیین می‌شود.
- بازه مجاز نمونه‌برداری 5 تا 20 ثانیه است.
- اگر مقدار نامعتبر بود → پیام خطا.
- اگر معتبر بود → مقدار جدید در \$cfg ذخیره شده و با save\_audio\_cfg در فایل نوشته می‌شود.

## بخش HTML ظاهر فرانت‌اند

## هدر

```
<!doctype html>
<html lang="fa" dir="rtl">
<head> ... </head>
```

- سند HTML با زبان فارسی و راست‌چین.
- کتابخانه‌های Bootstrap و آیکون‌ها بارگذاری می‌شوند.

## بدنه

```
<body class="bg-pattern">
<div class="container py-4">
```

- بدنه با یک بک‌گراند مشخص (bg-pattern).
- محتوای صفحه وسط‌چین می‌شود.

## پیام‌ها

```
<?php if($ok): ?>
 <div class="alert alert-success"><?=htmlspecialchars($ok)?></div>
<?php endif; ?>
```

- اگر \$ok پر باشد → پیام موفقیت سبز نمایش داده می‌شود.
- اگر \$err پر باشد → پیام خطا قرمز نمایش داده می‌شود.

## فرم تنظیمات

```
<form method="post" class="vstack gap-3">
```

- یک فرم POST با.
- دو بخش دارد:

1. گزینه استفاده از **ML** → checkbox روش/خاموش

2. انتخاب زمان نمونهبرداری 5 تا 20 ثانیه

#### دکمه‌ها

```
<div class="d-flex gap-2">
 <button class="btn btn-primary">ذخیره تنظیمات</button>
 بازگشت
</div>
```

- دکمه ذخیره تنظیمات.
- لینک بازگشت به داشبورد.

#### نمایش وضعیت فعلی

```

 <?= $cfg['use_ml'] ? 'ML' : 'ساده' ?>: حالت
 <?= intval($cfg['sample_sec']) ?>: زمان نمونهبرداری
 <code><?= htmlspecialchars(AUDIO_CFG_FILE) ?></code>

```

- بعد از فرم، وضعیت فعلی (حالت، زمان، مسیر فایل تنظیمات) نمایش داده می‌شود.

#### خلاصه عملکرد

این صفحه یک پنل تنظیمات تشخیص صدا است:

- کاربر می‌تواند انتخاب کند تشخیص صدا با **ML** انجام شود یا ساده.
- کاربر می‌تواند طول نمونهبرداری (5 تا 20 ثانیه) را انتخاب کند.

- بعد از ذخیره، مقادیر در فایل JSON آدرسش در AUDIO\_CFG\_FILE نوشته می‌شوند.
- پیام موفقیت یا خطا نمایش داده می‌شود.
- در پایین، وضعیت فعلی تنظیمات نشان داده می‌شود.

### Config.php:

```
<?php

date_default_timezone_set('Asia/Tehran');

ini_set('session.cookie_httponly', '1');

ini_set('session.use_strict_mode', '1');

session_name('IOTDASHSESSID');

define('STORAGE_PATH', __DIR__.'/storage');

if (!is_dir(STORAGE_PATH)) { @mkdir(STORAGE_PATH, 0755, true); }

define('USERS_FILE', STORAGE_PATH.'/users.json');
```

توضیح:

1

date\_default\_timezone\_set('Asia/Tehran');

- منطقه زمانی (Timezone) پیش‌فرض اسکریپت روی تهران تنظیم می‌شود.
- یعنی هر وقت با تاریخ (date(), time(), DateTime) کار شود، بر اساس ساعت ایران خواهد بود.

2

ini\_set('session.cookie\_httponly', '1');

- یکی از تنظیمات امنیتی برای سشن است.

وقتی **HttpOnly** فعال باشد، کوکی سشن با جاوااسکریپت در مرورگر قابل دسترسی نیست.

این کار جلوی حملات **XSS (Cross Site Scripting)** را می‌گیرد.

3

```
ini_set('session.use_strict_mode', '1');
```

این هم یک گزینه امنیتی برای سشن است.

وقتی فعال باشد، **PHP Session ID** معتبر و ایجادشده توسط خودش استفاده می‌کند.

اگر کاربر **Session ID** جعلی بفرستد، نادیده گرفته می‌شود و یک سشن جدید ساخته می‌شود.

4

```
session_name('IOTDASHSESSID');
```

نام پیشفرض سشن در PHP معمولاً **PHPSESSID** است.

این خط نام سشن را تغییر می‌دهد به **IOTDASHSESSID**.

این کار کمک می‌کند چند اپلیکیشن مختلف روی یک دامنه تداخل نداشته باشند.

5

```
define('STORAGE_PATH', __DIR__.'/storage');
```

یک ثابت تعریف می‌شود به اسم **STORAGE\_PATH**.

مقدار آن مسیر پوشش **storage** داخل همین فolder است = **DIR\_\_storage** مسیر همین فایل

یعنی هر داده ذخیره‌ای باید در این پوشش قرار بگیرد.

6

```
if (!is_dir(STORAGE_PATH)) { @mkdir(STORAGE_PATH, 0755, true); }
```

بررسی می‌کند اگر پوشش **storage** وجود ندارد (**!is\_dir(storage)**)، آن را بسازد (**mkdir()**).

• سطح دسترسی 0755 یعنی:

○ صاحب: خواندن + نوشن + اجرا

○ بقیه: فقط خواندن + اجرا

• یعنی پوشش‌های والد هم اگر لازم باشد ساخته شوند.

• @ یعنی خطای احتمالی (مثل نداشتن دسترسی) را مخفی کند.

7

```
define('USERS_FILE', STORAGE_PATH.'users.json');
```

• یک ثابت دیگر تعریف می‌شود.

• مسیر فایل کاربران : storage/users.json

• یعنی اطلاعات کاربرها (یوزرنیم/پسورد و غیره) به صورت JSON در این فایل نگه داشته می‌شود.

خلاصه:

این قطعه کد:

1. زمان را بر اساس تهران تنظیم می‌کند.

2. امنیت سشن‌ها را بالاتر می‌برد. (HttpOnly + Strict Mode).

3. یک نام اختصاصی برای سشن می‌گذارد. (IOTDASHSESSID)

4. مسیر ذخیره‌سازی (/storage) را مشخص می‌کند و اگر نبود، می‌سازد.

5. یک فایل JSON برای کاربران (users.json) تعریف می‌کند.

Dashboard.php:

```
<?php include '_top.php'; ?>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="assets/style.css?v=2">
```

```
<!-- اگر meta viewport آن را اضافه می کند در top.php نبود، این اسکریپت آن را اضافه می کند -->
```

```
<script>
```

```
(function(){
```

```
 if(!document.querySelector('meta[name="viewport"]')){
```

```
 var m=document.createElement('meta');
```

```
 m.name='viewport'; m.content='width=device-width, initial-scale=1, viewport-fit=cover';
```

```
 document.head.appendChild(m);
```

```
}
```

```
}());
```

```
</script>
```

```
<style>
```

```
/* تضمین تکستونه شدن در موبایل حتی اگر استایل‌های دیگر دخالت کنند */
```

```
@media (max-width: 575.98px){
```

```
 .stack-xs > .col, .stack-xs > [class^="col"] { flex: 0 0 100% !important; max-width: 100% !important; }
```

```
 .card .btn { width: 100%; }
```

```
 .card .card-body { padding: 12px 14px; }
```

```
}
```

```
</style>
```

```
<div class="container py-4">
```

```
<!-- KPI 1s -->

<div class="row row-cols-2 row-cols-md-4 g-3">
 <div class="col">
 <div class="card kpi shadow-sm h-100">
 <div class="card-body">
 <div class="kpi-label">CPU</div>
 <div class="kpi-value" id="cpu">--%</div>
 </div>
 </div>
 </div>
 <div class="col">
 <div class="card kpi shadow-sm h-100">
 <div class="card-body">
 <div class="kpi-label">RAM</div>
 <div class="kpi-value" id="ram">--%</div>
 </div>
 </div>
 </div>
 <div class="col">
 <div class="card kpi shadow-sm h-100">
 <div class="card-body">
 <div class="kpi-label">Temp</div>
 <div class="kpi-value" id="temp">--°C</div>
 </div>
 </div>
 </div>
 </div>
</div>
```

```
</div>

<div class="col">
 <div class="card kpi shadow-sm h-100">
 <div class="card-body">
 <div class="kpi-label">IP</div>
 <div class="kpi-sub" id="net">—</div>
 </div>
 </div>
</div>

</div>
```

!-- کارت‌های اصلی : ردیف ۱ -->

```
<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-2 g-md-3 mt-1 align-items-stretch stack-xs">
```

```
<div class="col">
 <div class="card shadow-sm h-100">
 <div class="card-body d-flex flex-column">
 <div class="d-flex align-items-center mb-2">
 <i class="bi bi-mic fs-4 me-2"></i>
 <h6 class="mb-0 fw-bold">رویدادهای صوتی</h6>
 </div>
 <p class="card-desc mb-3">مشاهده و فیلتر آخرین رویدادهای صوتی ضبط شده</p>
 مشاهده رویدادهای صوتی
 </div>
 </div>
```

```

</div>

<div class="col">
 <div class="card shadow-sm h-100">
 <div class="card-body d-flex flex-column">
 <div class="d-flex align-items-center mb-2">
 <i class="bi bi-activity fs-4 me-2"></i>
 <h6 class="mb-0 fw-bold">پردازهای</h6>
 </div>
 <p class="card-desc mb-3">فهرست پردازهای فعال با مصرف CPU/RAM.</p>
 مشاهده پردازهای
 </div>
 </div>
</div>

<div class="col">
 <div class="card shadow-sm h-100">
 <div class="card-body d-flex flex-column">
 <div class="d-flex align-items-center mb-2">
 <i class="bi bi-graph-up-arrow fs-4 me-2"></i>
 <h6 class="mb-0 fw-bold">متريکها</h6>
 </div>
 <p class="card-desc mb-3">روند CPU/RAM و خلاصه وضعیت شبکه</p>
 مشاهده متريکها
 </div>
 </div>

```

```

</div>
</div>

</div>

--> ردیف ۲: تنظیمات تشخیص صدا + محدودسازی رویدادها-->

<div class="row row-cols-1 row-cols-md-2 g-2 g-md-3 mt-1 stack-xs">

<div class="col">
 <div class="card shadow-sm h-100">
 <div class="card-body d-flex flex-column">
 <div class="d-flex align-items-center mb-2">
 <i class="bi bi-sliders2 fs-4 me-2"></i>
 <h6 class="mb-0 fw-bold">تنظیمات تشخیص صدا</h6>
 </div>
 <p class="card-desc mb-3">یا ساده و تعیین پازه‌ی نمونه‌برداری (۵ تا ۲۰ ثانیه) ML انتخاب حالت</p>
 باز کردن تنظیمات
 </div>
 </div>
</div>

<div class="col">
 <div class="card shadow-sm h-100">
 <div class="card-body d-flex flex-column">

```

```

<div class="d-flex align-items-center mb-2">
 <i class="bi bi-x-octagon fs-4 me-2"></i>
 <h6 class="mb-0 fw-bold">محدودسازی رویدادها در تشخیص صدا</h6>
</div>

<p class="card-desc mb-3">موارد تیک خورده در تشخیص صوت نادیده گرفته می‌شوند</p>

تنظیم محدودسازی
</div>
</div>
</div>

</div>

```

```

</div>

<script src="assets/app.js"></script>
<script>IOT.initDashboard();</script>

```

توضیح :

**1**

<?php include '\_top.php'; ?>

- فایل top.php لود می‌شود.

- معمولاً این فایل شامل هدر مشترک، سشن، احرار هویت یا تنظیمات متأ است.

**2**

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="assets/style.css?v=2">
```

- اضافه کردن **Bootstrap CSS** برای استایل‌ها.

- اضافه کردن **Bootstrap Icons** برای آیکون‌ها.

- فایل استایل اختصاصی پروژه (style.css).

3

```
<script>
```

```
(function(){
```

```
if(!document.querySelector('meta[name="viewport"]')){
```

```
var m=document.createElement('meta');
```

```
m.name='viewport'; m.content='width=device-width, initial-scale=1, viewport-fit=cover';
```

```
document.head.appendChild(m);
```

```
}
```

```
}());
```

```
</script>
```

- این اسکریپت بررسی می‌کند اگر **meta viewport** وجود نداشت، خودش اضافه کند.

- **viewport** برای ریسپانسیو شدن صفحه در موبایل ضروری است.

4

```
<style>
```

```
@media (max-width: 575.98px){
```

```
.stack-xs > .col, .stack-xs > [class^="col"] { flex: 0 0 100% !important; max-width: 100% !important; }
```

```

.card .btn { width: 100%; }

.card .card-body { padding: 12px 14px; }

}

</style>

```

- استایل اختصاصی برای موبایل صفحه‌های کوچک‌تر از 576 px
- همه کارت‌ها تک‌ستونه شوند.
- دکمه‌ها عرض کامل داشته باشند.
- فاصله داخل کارت‌ها کم‌تر شود.

## بخش محتوای داشبورد

### کانتینر اصلی

```
<div class="container py-4">
```

- یک کانتینر Bootstrap با padding بالا و پایین.

### KPI‌ها (شاخص‌های کلیدی)

```
<div class="row row-cols-2 row-cols-md-4 g-3">
```

- IP، RAM، CPU، دما، رديفی با ۴ کارت در موبایل ۲ تا در هر رديف → در دسکتاپ ۴ تایی.
- هر کارت:

- عنوان مثلث CPU
- مقدار پیش‌فرض  ${}^{\circ}\text{C}$  یا  ${}^{\circ}\text{C}$  یا  $-\text{--}^{\circ}\text{C}$ .
- جاهایی مثل `<div id="cpu">` بعداً با جاوا اسکریپت پر می‌شوند.

## ردیف ۱: کارت‌های اصلی

```
<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-2 g-md-3 mt-1 align-items-stretch stack-xs">
```

کارت‌های اصلی داشبورد:

1. رویدادهای صوتی → لینک به events.php.

2. پردازه‌ها → لینک به processes.php.

3. متريک‌ها → لینک به metrics.php.

هر کارت: آيکون + عنوان + توضیح کوتاه + دکمه.

طراحی طوری است که ارتفاع همه کارت‌ها برابر (h-100) باشد.

## ردیف ۲: تنظیمات صوتی

```
<div class="row row-cols-1 row-cols-md-2 g-2 g-md-3 mt-1 stack-xs">
```

دو کارت دیگر:

1. تنظیمات تشخیص صدا → لینک به audio\_settings.php.

2. محدودسازی رویدادها → لینک به audio\_blocklist.php.

این کارت‌ها امکان تنظیم حالت ML و فیلتر کردن صدای خاص را می‌دهند.

## بخش اسکریپت‌ها

1

```
<script src="assets/app.js"></script>
```

فایل جاوا اسکریپت اصلی پروژه لود می‌شود.

2

```
<script>IOT.initDashboard();</script>
```

تابع IOT.initDashboard() اجرا می‌شود.

- این تابع احتمالاً در app.js تعریف شده و وظیفه دارد:
- دریافت مقادیر CPU/RAM/Temp/IP از سرور یا MQTT.
- آپدیت کردن المان‌هایی مثل #cpu, #ram, #temp, #net.

## جمع‌بندی

این فایل در واقع صفحه داشبورد اصلی IoT است:

- بالای صفحه KPI‌های سریع.
- ردیف ۱: لینک سریع به رویدادهای صوتی، پردازه‌ها، متريک‌ها.
- ردیف ۲: تنظیمات مربوط به تشخیص صوت، ML بازه نمونه‌برداری و محدودسازی صداها.
- همه کارت‌ها واکنش‌گرا هستند و در موبایل تکستونه نمایش داده می‌شوند.
- داده‌ها به صورت زنده توسط app.js بارگذاری و بروزرسانی می‌شوند.

### Db.php:

```
<?php
$cfg = require __DIR__.'/env.php';

$mysqli = new mysqli($cfg['DB_HOST'],$cfg['DB_USER'],$cfg['DB_PASS'],$cfg['DB_NAME']);

if ($mysqli->connect_errno) { http_response_code(500); die("DB error"); }

$mysqli->set_charset("utf8mb4");
```

توضیح :

```
$cfg = require __DIR__.'/env.php';
```

فایل env.php که حاوی تنظیمات (مثل نام دیتابیس، کاربر، رمز و ...) هست لود می‌شه.

این فایل باید یک آرایه PHP برگردانه. مثلاً چیزی شبیه:

```
?>php
return]
' DB_HOST' => 'localhost,'
' DB_USER' => 'iotapp,'
' DB_PASS' => 'ChangeMe_Strong,!'
' DB_NAME' => 'iotdash,'
;[
```

متغیر \$cfg همین آرایه رو می‌گیره.

```
$mysqli = new mysqli($cfg['DB_HOST'],$cfg['DB_USER'],$cfg['DB_PASS'],$cfg['DB_NAME']);
```

یک اتصال جدید به MySQL ساخته میشے با کلاس `.mysqli`

ورودی‌ها از فایل کانفیگ گرفته میشے (`.host, user, pass, db`)

```
if ($mysqli->connect_errno) { http_response_code(500); die("DB error"); }
```

چک می‌کنه که آیا اتصال موفق بوده یا خطأ داده.

اگر خطأ (connect\_errno) داشت:

کد پاسخ HTTP رو 500 (خطای سرور) میذاره.

با ("die("DB error")) اجرای اسکریپت رو متوقف می‌کنه و پیام خطأ می‌نویسه.

```
$mysqli->set_charset("utf8mb4");
```

مجموعه کاراکتر اتصال رو روی utf8mb4 میداره (بهترین حالت برای ساپورت کامل یونیکد، مثل اموجی و حروف فارسی).

خلاصه: این اسکریپت یک اتصال امن و استاندارد به دیتابیس MySQL برقرار می‌کنه با تنظیمات در .env.php، خطا رو به‌شکل درست هندل می‌کنه، و کاراکترست رو روی UTF-8 کامل تنظیم می‌کنه.

### Events.php:

```
<?php include '_top.php'; ?>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css">

<div class="container py-4">

<!-- فیلترها -->

<div class="d-flex align-items-end gap-2 mb-3 flex-wrap">

<div>
 <label class="form-label">ز</label>
 <input id="start" type="datetime-local" class="form-control form-control-sm">
</div>

<div>
 <label class="form-label">تا</label>
 <input id="end" type="datetime-local" class="form-control form-control-sm">
</div>

<div class="flex-grow-1">
 <label class="form-label">جستجو</label>
</div>
```

```

<input id="search" class="form-control form-control-sm" placeholder="برچسب / گره">
</div>

<!-- فیلترهای جدید: کانفیدنس -->
<div>
 <label class="form-label">(%) حداقل اعتماد</label>
 <input id="confMin" type="number" min="0" max="100" step="0.1" class="form-control form-control-sm" placeholder="60 مثلاً">
</div>
<div>
 <label class="form-label">(%) حداکثر اعتماد</label>
 <input id="confMax" type="number" min="0" max="100" step="0.1" class="form-control form-control-sm" placeholder="95 مثلاً">
</div>

<button class="btn btn-sm btn-primary btn-filter-fixed" onclick="IOT.applyEventFilters()">
 <i class="bi bi-funnel"></i> اعمال فیلتر
</button>
</div>

<!-- جدول -->
<div class="card shadow-sm">
 <div class="card-body p-0">
 <div class="table-responsive">
 <table class="table table-striped table-sm mb-0">
</pre>

```

```

<thead class="table-light">
 <tr>
 <th>#</th>
 <th>زمان</th>
 <th>برچسب</th>
 <th>اعتماد</th>
 <th>هجرگ</th>
 </tr>
</thead>
<tbody id="eventsFull"></tbody>
</table>
</div>
</div>
</div>

<!-- صفحه‌بندی + وضعیت جست‌وجو -->
<div class="d-flex align-items-center justify-content-between mt-3 flex-wrap gap-2">
 <div class="d-flex align-items-center gap-2">
 <label class="small ev-status">تعداد در هر صفحه:</label>
 <select id="evPer" class="form-select form-select-sm" style="width:auto">
 <option value="25">25</option>
 <option value="50" selected>50</option>
 <option value="100">100</option>
 </select>
 </div>
</div>

```

```

<nav>
 <ul class="pagination pagination-sm mb-0">
 <li class="page-item"><button id="evPrev" class="page-link"
 type="button">«</button>
 <li class="page-item disabled">
 صفحة 1 1

 <li class="page-item"><button id="evNext" class="page-link"
 type="button">»</button>

</nav>

<div class="small ev-status">
 سقف نمایش: 0 (1000000)
 · —
</div>
</div>

<script src="assets/app.js"></script>
<script>
 IOT.initEventsPage();
</script>

```

توضیح :  
سربرگ و استایل‌ها

```
<?php include '_top.php'; ?>
```

- بارگذاری هدر/سشن/امنیت و هر چیزی که در `<head>` دارید (مثل `top.php` مشترک، منو، چک لاین)

```
<link ... bootstrap.min.css>
```

```
<link ... bootstrap-icons.css>
```

```
<link rel="stylesheet" href="assets/style.css">
```

- بوت استرپ برای گرید و استایل، آیکون‌های بوت استرپ، و استایل اختصاصی پروژه.

بدنه صفحه و فیلترها

```
<div class="container py-4">
```

- ظرف اصلی با فاصله عمودی.

```
<div class="d-flex align-items-end gap-2 mb-3 flex-wrap">
```

- ردیف کنترل‌ها: چیدمان فلکسیبل، انتهای هم‌تراز، قابل‌بسته‌بندی در موبایل.

بازه زمانی

```
<label>از</label>
```

```
<input id="start" type="datetime-local" class="form-control form-control-sm">
```

- ورودی تاریخ/زمان شروع (HTML5). مقدار با JS خوانده می‌شود.

```
<label>تا</label>
```

```
<input id="end" type="datetime-local" class="form-control form-control-sm">
```

- ورودی تاریخ/زمان پایان.

جست‌وجوی متنی

```
<label>جست‌وجو</label>
```

```
<input id="search" class="form-control form-control-sm" placeholder="برچسب / گره">
```

- فیلتر متن آزاد: معمولاً روی ستون‌های «برچسب» یا «گره» اعمال می‌شود.

### (Confidence) فیلترهای اعتماد

<label> (%) حداقل اعتماد</label>

<input id="confMin" type="number" min="0" max="100" step="0.1" ... placeholder="مثالاً 60">

- حداقل درصد اعتماد (۰ تا ۱۰۰ با گام ۰.۱).

<label> (%) حداکثر اعتماد</label>

<input id="confMax" type="number" min="0" max="100" step="0.1" ... placeholder="مثالاً 95">

- حداکثر درصد اعتماد.

### دکمه اعمال فیلتر

<button ... onclick="IOT.applyEventFilters()">

اعمال فیلتر

</button>

- با کلیک، تابع (IOT.applyEventFilters()) اجرا می‌شود تا داده‌ها با پارامترهای فعلی فیلتر/واکنشی شوند.

### جدول رویدادها

<div class="card shadow-sm">

<div class="table-responsive">

<table class="table table-striped table-sm mb-0">

<thead class="table-light">

<tr>

<th>#</th>

<th>زمان</th>

```

<th>برچسب</th>
<th>اعتماد</th>
<th>گره</th>
</tr>
</thead>
<tbody id="eventsFull"></tbody>
</table>
</div>
</div>

```

کارت حاوی جدول واکنش‌گرا. •

ستون‌ها: •

◦ شمارنده ردیف در همین صفحه. ◦

◦ زمان timestamp رویداد، معمولاً با فرمت محلی/قابل خواندن

◦ برچسب label اطیقه‌بندی صوت

◦ اعتقاد confidence درصدی

◦ گره node\_id یا منبع

بدنه جدول tbody#eventsFull توسط جاوا اسکریپت پر می‌شود. •

### صفحه‌بندی و وضعیت

```
<select id="evPer"> ... </select>
```

«تعداد در هر صفحه»: 100/50/25 (پیش‌فرض 50). تغییرش باید رفرش داده‌ها را با اندازه صفحه جدید تریگر کند. •

```
<ul class="pagination pagination-sm">
```

```
<button id="evPrev"><</button>
```

```

<li class="disabled">صفحه 11
<button id="evNext">»</button>

```

• </ul>

کنترل صفحه‌بندی ساده:

◦ #evPrev رفتن به صفحه قبل؛

◦ #evNext صفحه بعد؛

◦ #evPage شماره صفحه جاری؛

◦ #evPages کل صفحات.

• دکمه‌ها در JS باید بر اساس وضعیت ابتدای/انتهای لیست غیرفعال/فعال شوند.

```
<div class="small ev-status">
```

(مجموع: <span id="evTotal">0</span> (1000000) نمایش سقف:

◦ . <span id="evPerf">—</span>

```
</div>
```

◦ #evTotal کل نتایج فیلترشده.

◦ سقف نمایش (limit) سمت سرور/کلاینت (برای جلوگیری از بارگیری بسیار سنگین).

◦ #evPerf نمایش زمان اجرای کوئری/واکشی (برای دیباگ و تجربه کاربری).

## اسکریپت‌ها و راهاندازی

```
<script src="assets/app.js"></script>
```

```
<script>
```

```
IOT.initEventsPage();
```

```
</script>
```

◦ فایل JS اصلی.

• **ابايد: OT.initEventsPage()**

- لیستنرها را روی فیلترها/صفحه‌بندی ثبت کند،
- مقدار پیش‌فرض evPer را بخواند،
- بر اساس start/end/search/confMin/confMax/page/size خواست داده بفرستد،
- جواب را در eventsFull رندر کند،
- #evTotal, #evPage, #evPages, #evPerf را به‌روزرسانی کند.

نکات کاربردی /امنیتی /تجربی

- ورودی‌ها را سمت سرور هم اعتبارسنجی کنیم بازه تاریخ، min/max اعتماد، طول جست‌و‌جو
- اگر ISO datetime-local با فرمت محلی سازگار نیست، در سمت کلاینت مقدار را به (UTC) تبدیل کنیم و همراه timezone یا سمت سرور را بر مبنای Asia/Tehran ثابت مثلاً offset/Zone تفسیر کنیم
- برای «جست‌و‌جو»، اگر قرار است هم برچسب و هم گره را پوشش دهد، در API پارامتر را به دو فیلد map کنیم مثلاً bindlabel LIKE ? OR node\_id LIKE ?
- برای «اعتماد»، بهتر است ورودی‌های خالی را نادیده بگیریم و فقط وقتی مقدار داده شده فیلتر کنیم. همچنان اگر confMin > confMax بود، خطای نشان دهیم
- حتماً در پاسخ API، داده‌ها را paginate کنیم و تعداد کل (total) را برگردانیم تا #evTotal و #evPages را انجام میدهیم
- درست محاسبه شوند.
- اگر داده زیاد است، ایندکس‌های مناسب روی ستون‌های زمان/برچسب/گره/اعتماد بگذاریم که این کار انجام میدهیم

Index.php:

```
<?php
session_start();

require_once __DIR__.'/config.php';

/**
```

\* گذروازه >= کاربران مجاز (نام کاربری)

\* در صورت نیاز کاربران بیشتری اضافه کنید —

\*/

```
$valid_users = [
 'Ali' => 'EmbeddedProject',
 'Hassan' => '1234',
];
```

/\*\* پیکربندی محدودیت تلاش‌ها \*/

```
const MAX_ATTEMPTS = 4; // حداقل تلاش ناموفق
const LOCK_SECONDS = 10; // مدت قفل شدن (ثانیه)
```

/\*\* JSON خواندن تاریخ آخرین ورود کاربران از فایل \*/

```
function load_users() {
 if (file_exists(USER_FILE)) {
 $j = json_decode(file_get_contents(USER_FILE), true);
 if (is_array($j)) return $j;
 }
 return [];
}
```

/\*\* JSON ذخیره تاریخ آخرین ورود در فایل \*/

```
function save_users($arr) {
 @file_put_contents(USER_FILE, json_encode($arr,
 JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT));
```

```
}
```

```
/*+ IP) بر اساس نام کاربری) کلید یکتا برای مدیریت قفل/تلashها */
```

```
function login_key(string $username): string {
 $ip = $_SERVER['REMOTE_ADDR'] ?? '0.0.0.0';
 return strtolower(trim($username)) . '@' . $ip;
}
```

```
/* آمده سازی ساختار سشن برای لگین */
```

```
if (!isset($_SESSION['login'])) {
 $_SESSION['login'] = [
 'attempts' => [], // [key => count]
 'locks' => [], // [key => timestamp]
];
}
```

```
$err = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $u = trim($_POST['username'] ?? "");
 $p = (string)($_POST['password'] ?? "");
 $now = time();
 $key = login_key($u);
```

```
// بررسی قفل بودن
```

```
$lockUntil = $_SESSION['login']['locks'][$key] ?? 0;
```

```

if ($now < $lockUntil) {

$wait = $lockUntil - $now;

$err = "ثانیه صبر کنید {$wait} به دلیل تلاش‌های ناموفق، ورود موقتاً مسدود است. لطفاً";

} else {

 اگر از قبل قفل گذشته، مطمئن شو پاک شده //

 if ($lockUntil > 0 {

 unset($_SESSION['login']['locks'][$key]);

 }

 اعتبارسنجی کاربر //

 if ($u !== "" && isset($valid_users[$u]) && hash_equals($valid_users[$u], $p)) {

 موفقیت: ریست شمارندها و قفل //

 unset($_SESSION['login']['attempts'][$key], $_SESSION['login']['locks'][$key]);

 ثبت و بروزرسانی آخرین ورود //

 $users = load_users();

 $last = $users[$u]['last_login_at'] ?? null;

 $_SESSION['uid'] = crc32($u); // شناسه یکتا بر اساس نام کاربری

 $_SESSION['username'] = $u;

 $_SESSION['last_login_prev'] = $last;

 $users[$u] = ['last_login_at' => date('Y-m-d H:i:s')];

 save_users($users);
}

```

```

header('Location: dashboard.php');

exit;

} else {

 خطای ورود // افزایش شمارنده تلاش برای این key

 $cnt = (int)($_SESSION['login']['attempts'][$key] ?? 0);

 $cnt++;

 $_SESSION['login']['attempts'][$key] = $cnt;

 if ($cnt >= MAX_ATTEMPTS) {

 قفل ۱۰ ثانیه‌ای //

 $_SESSION['login']['locks'][$key] = $now + LOCK_SECONDS;

 ریست شمارنده تلاش‌ها تا بعد از اتمام قفل دوباره از + شروع شود //

 unset($_SESSION['login']['attempts'][$key]);

 $err = "ثانیه صبر کنید و سپس دوباره تلاش کنید.".LOCK_SECONDS." بیش از حد تلاش ناموفق. لطفاً";

 } else {

 $remain = MAX_ATTEMPTS - $cnt;

 $err = "نام کاربری یا گذرواژه نادرست است. (تعداد تلاش باقیمانده {$remain})";

 }

}

?>

<!doctype html>

<html lang="fa" dir="rtl">

```

```

<head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>ورود | IoT Dash</title>
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
 <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
 <link rel="stylesheet" href="assets/style.css?v=2">
</head>
<body class="bg-pattern d-flex align-items-center" style="min-height:100vh;">
 <div class="container">
 <div class="row justify-content-center">
 <div class="col-lg-4 col-md-6">
 <div class="card card-glass shadow-lg">
 <div class="card-body p-4">
 <div class="text-center mb-3">
 <i class="bi bi-cpu fs-1 text-primary"></i>
 <h4 class="mt-2 login-title">ورود به پنل IoT</h4>
 <p class="text-muted small mb-0">Orange Pi • LAMP • Audio Events</p>
 </div>
<?php if($err): ?>
 <div class="alert alert-danger py-2"><?=htmlspecialchars($err)?></div>
<?php endif; ?>
 </div>
 </div>
 </div>
 </div>
</body>

```

```

<form method="post" class="needs-validation" novalidate>

 <div class="mb-3">
 <label class="form-label">نام کاربری</label>
 <input name="username" class="form-control" placeholder="نام کاربری" required>
 <div class="invalid-feedback">نام کاربری را وارد کنید</div>
 </div>

 <div class="mb-3">
 <label class="form-label">گذرواژه</label>
 <input name="password" type="password" class="form-control" placeholder="گذرواژه" required>
 <div class="invalid-feedback">گذرواژه را وارد کنید.</div>
 </div>

 <button class="btn btn-primary w-100" type="submit">ورود</button>
</form>
</div>

<div class="card-footer text-center small text-muted py-2">
 © IoT Dash
</div>
</div>
</div>
</div>

<script>

```

```

document.getElementById('year').textContent = new Date().getFullYear();

// اعتبارسنجی سمت کاربر (Bootstrap)
(() => {
 'use strict';

 const forms = document.querySelectorAll('.needs-validation');

 Array.from(forms).forEach(form => {
 form.addEventListener('submit', e => {
 if (!form.checkValidity()) { e.preventDefault(); e.stopPropagation(); }

 form.classList.add('was-validated');
 }, false);
 });
})();

```

</script>

</body>

</html>

توضیح :

**بخش PHP منطقی**

اشروع سشن و بارگذاری تنظیمات

session\_start();

require\_once \_\_DIR\_\_.'/config.php';

• session\_start() را برای نگهداری وضعیت کاربر (مثل تلاش‌ها، قفل شدن، نام کاربری) فعال می‌کند.

• config.php را می‌آورد (معمولاً تنظیمات عمومی سایت/استایل/هدر و...).

## ۲ تعریف کاربران مجاز

```
$valid_users = [
 'Ali' => 'EmbeddedProject',
 'Hassan' => '1234',
];
```

- یک دیکشنری «نام کاربری ⇒ گذرواژه» برای لایگین.

نکته امنیتی: بهتر است در عمل از پسورد هش شده در DB استفاده شود نه متن ساده

## ۳ پیکربندی محدودیت تلاش‌ها

```
const MAX_ATTEMPTS = 4; // حداکثر تلاش ناموفق
const LOCK_SECONDS = 10; // مدت قفل (ثانیه)
```

- اگر تلاش‌های ناموفق برای یک کاربر در یک IP به ۴ رسید، ۱۰ ثانیه قفل می‌شود.

## ۴ خواندن/نوشتن تاریخ آخرین ورود از فایل JSON

```
function load_users() {
 if (file_exists(USER_FILE)) {
 $j = json_decode(file_get_contents(USER_FILE), true);
 if (is_array($j)) return $j;
 }
 return [];
}

function save_users($arr) {
 @file_put_contents(USER_FILE, json_encode($arr,
 JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT));
}
```

```
}
```

- اگر `load_users()` وجود داشته باشد، JSON را به آرایه تبدیل و بر می گرداند؛ و گرنه آرایه خالی.
- آرایه را به JSON خوانا) یونیکد (PRETTY + تبدیل و ذخیره می کند.
- فرض: ثابت `define('USERS_FILE', config.php` تعريف شده است؛ مثلًا، `__DIR__.'/config/users.json');`

## ۵ کلید یکتا برای مدیریت تلاش ها / قفل

```
function login_key(string $username): string {
 $ip = $_SERVER['REMOTE_ADDR'] ?? '0.0.0.0';
 return strtolower(trim($username)).'@'.$ip;
}
```

- برای هر ترکیب «نام کاربری «IP + یک کلید می سازد؛ این باعث می شود حملات از IP های مختلف، مستقل شمرده شوند و کاربر واقعی از IP خودش قفل شود، نه جهانی.

## ۶ آماده سازی فضای سشن برای لاغین

```
if (!isset($_SESSION['login'])) {
 $_SESSION['login'] = [
 'attempts' => [], // [key => count]
 'locks' => [], // [key => timestamp]
];
}
```

- اگر ساختار `$_SESSION['login']` وجود ندارد، می سازد:

تعداد تلاش های ناموفق تا الان برای هر کلید.

زمان تا وقتی که اجازه هی تلاش مجدد نیست (تایم استمپ یونیکس).

## ۷ پردازش در خواست POST

```
$err = "";

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

 $u = trim($_POST['username'] ?? "");
 $p = (string)($_POST['password'] ?? "");

 $now = time();
 $key = login_key($u);
```

- فقط وقتی فرم ارسال شده (POST) وارد فرآیند می‌شود.

username و password خوانده می‌شوند نام کاربری trim، رمز به رشتہ

- زمان فعلی (\$now) و کلید یکتا (\$key) محاسبه می‌شود.

## ۸ بررسی قفل بودن کاربر @IP

```
$lockUntil = $_SESSION['login']['locks'][$key] ?? 0;
if ($now < $lockUntil) {
```

\$wait = \$lockUntil - \$now;

\$err = " به دلیل تلاش‌های ناموفق، ورود مؤقتاً مسدود است. لطفاً {\$wait} ثانیه صبر کنید;".

- اگر هنوز در بازه‌ی قفل هست، پیام می‌دهد چند ثانیه دیگر صبر کنید و لاغرین انجام نمی‌شود.

## ۹ اگر قفل گذشته باشد، پاکسازی قفل

```
} else {
 if ($lockUntil > 0) {
 unset($_SESSION['login']['locks'][$key]);
 }
```

- اگر قفل قبلی تمام شده، اثرش از سشن پاک می‌شود.

## ۱۰ اعتبارسنجی کاربر/رمز

```
if ($u !== "" && isset($valid_users[$u]) && hash_equals($valid_users[$u], $p)) {
```

سه شرط:

- نام کاربری خالی نباشد.
- در آرایه کاربران تعریف شده باشد.
- مقدار گذرواره با `hash_equals` مقایسه شود (مقایسه امن برای جلوگیری از حملات زمان‌بندی؛ هرچند اینجا پسوردها «هش» نیستند، اما مقایسه ثابت‌زمانی است).

## ۱۱ الگین موفق: ریست تلاش‌ها/قفل، ثبت زمان

```
unset($_SESSION['login']['attempts'][$key], $_SESSION['login']['locks'][$key]);
```

```
$users = load_users();
$last = $users[$u]['last_login_at'] ?? null;
```

```
$_SESSION['uid'] = crc32($u); // شناسه یکتا از نام کاربری
$_SESSION['username'] = $u;
$_SESSION['last_login_prev'] = $last;
```

```
$users[$u] = ['last_login_at' => date('Y-m-d H:i:s')];
save_users($users);
```

```
header('Location: dashboard.php');
exit;
```

شمارنده تلاش‌ها و قفل برای این کلید پاک می‌شود.

- فایل کاربران خوانده می‌شود تا «آخرین ورود قبلی» استخراج شود و داخل سشن گذاشته شود (last\_login\_prev) تا بالای داشبورد نشان دهید.
- سشن کاربر سنت می‌شود:

  - uid اساس بر اساس crc32 نام کاربری (یونیفای ساده برای داشتن یک شناسه‌ی عددی).
  - username هم برای نمایش.

- «آخرین ورود جدید» با زمان فعلی ذخیره می‌شود و سپس ریدایرکت به داشبورد.

## ۱۲. لاگین ناموفق: افزایش شمارنده، اعمال قفل یا اعلام باقی‌مانده

```

} else {

$cnt = (int)($_SESSION['login']['attempts'][$key] ?? 0);

$cnt++;

$_SESSION['login']['attempts'][$key] = $cnt;

if ($cnt >= MAX_ATTEMPTS) {

$_SESSION['login']['locks'][$key] = $now + LOCK_SECONDS;

unset($_SESSION['login']['attempts'][$key]);

$err = "ثانية صبر كنيد و سپس دوباره تلاش کنيد ".LOCK_SECONDS." . بيش از حد تلاش ناموفق. لطفاً";

}.ثانیه صبر کنید و سپس دوباره تلاش کنید ".LOCK_SECONDS." . بیش از حد تلاش ناموفق. لطفاً

} else {

$remain = MAX_ATTEMPTS - $cnt;

$err = "نام کاربری یا گذرواژه نادرست است. (تعداد تلاش باقی‌مانده {$remain})";

}

}

}

}

}

```

- هر بار خطای شمارنده تلاش‌های این کلید + ۱.

اگر به سقف رسید: قفل تا `$now + LOCK_SECONDS`، شمارنده ریست، پیام قفل.

اگر هنوز به سقف نرسیده: پیام «غلط است» + تعداد تلاش باقیمانده.

## بخش HTML رندر فرم

### ۱۳ اساختار پایه‌ی صفحه

```
<!doctype html>
<html lang="fa" dir="rtl">
<head> ... </head>
<body class="bg-pattern d-flex align-items-center" style="min-height:100vh;">
```

سند HTML با زبان فارسی و راستبهچپ.(`dir="rtl"`).

CSS خودت (`assets/style.css?v=2`) لود می‌شوند.

بدنه با پسزمنه‌ی الگو و همترازی عمودی برای مرکزکردن کارت.

### ۱۴ کارت لاغین

یک کارت شیشه‌ای (`card-glass`) وسط صفحه:

آیکون CPU ، تیتر «ورود به پنل» IoT ، زیرتیتر.

### ۱۵ نمایش خطأ (اگر باشد)

```
<?php if($err): ?>
<div class="alert alert-danger py-2"><?=htmlspecialchars($err)?></div>
<?php endif; ?>
```

اگر \$err مقدار دارد، بهصورت هشدار قرمز نمایش داده می‌شود برای امنیت XSS HTML-escaped

### ۱۶ فرم لاغین

```
<form method="post" class="needs-validation" novalidate>
```

```
<div class="mb-3">
```

```

<label>نام کاربری</label>

<input name="username" class="form-control" placeholder="نام کاربری required">

<div class="invalid-feedback">.نام کاربری را وارد کنید</div>

</div>

<div class="mb-3">

<label>گذرواژه</label>

<input name="password" type="password" class="form-control" placeholder="گذرواژه" required>

<div class="invalid-feedback">.گذرواژه را وارد کنید</div>

</div>

<button class="btn btn-primary w-100" type="submit">ورود</button>

</form>

```

فرم POST ساده با اعتبارسنجی سمت کاربر (Bootstrap) و پیام‌های خطای لازم.

## ۱۷ فوتر کارت

```

<div class="card-footer text-center small text-muted py-2">

© IoT Dash

</div>

```

سال جاری به صورت دینامیک درج می‌شود.

## بخش جواااسکریپت (پایین)

### ۱۸ درج سال جاری

```
document.getElementById('year').textContent = new Date().getFullYear();
```

## ۱۹ افعال‌سازی اعتبارسنجی Bootstrap

```
() => {
```

```

'use strict';

const forms = document.querySelectorAll('.needs-validation');

Array.from(forms).forEach(form => {

 form.addEventListener('submit', e => {

 if (!form.checkValidity()) { e.preventDefault(); e.stopPropagation(); }

 form.classList.add('was-validated');

 }, false);

});

})();

```

- مانع ارسال فرم می‌شود اگر فیلدهای `required` پر نشده باشند؛ کلاس `was-validated` برای نمایش پیام‌های خطای فرم.

### جمع‌بندی رفتار کلی

- هر کاربر از یک IP اگر ۴ بار پشت‌سرهم اشتباه بزند، برای ۱۰ ثانیه قفل می‌شود.
- در لاغین موفق:
  - سشن با `username/last_login_prev` شدت می‌شود،
  - آخرین ورود در فایل JSON ذخیره می‌گردد،
  - و کاربر به داشبورد می‌رود.
- در لاغین ناموفق:
  - تعداد تلاش‌ها شمرده می‌شود،
  - پیام مناسب (باقی‌مانده یا قفل) نمایش داده می‌شود.

### Logout.php:

```
<?php session_start(); session_destroy(); header('Location: index.php');
```

توضیح :

```
session_start();
```

سشن فعلی PHP را شروع می‌کند.

اگر سشن از قبل وجود داشته باشد، همان را بازیابی می‌کند.

برای اینکه بتوانیم سشن را نابود کنیم، باید اول آن را باز کنیم.

2

```
session_destroy();
```

تمام داده‌های سشن فعلی را پاک می‌کند.

به عبارتی کاربر خارج (logout) می‌شود و اطلاعات ورود دیگر ذخیره نمی‌ماند.

دقت کن: این دستور فقط داده‌های سمت سرور را پاک می‌کند، کوکی سشن در مرورگر هنوز باقی می‌ماند تا وقتی که expire شود (یا با کد جداگانه پاک شود).

3

```
header('Location: index.php');
```

بعد از پاک شدن سشن، کاربر به صفحه index.php هدایت (redirect) می‌شود.

معمولًاً index.php صفحه ورود (Login Page) یا صفحه اصلی بدون لاجین است.

خلاصه:

این فایل یک logout.php ساده است:

سشن باز می‌شود.

سشن نابود می‌شود.

کاربر به صفحه اصلی (index.php) برگردانده می‌شود.

#### Metrics.php:

```
<?php include '_top.php'; ?>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css">

<div class="container py-4">
 <div class="row g-3">
 <div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header">نمونه متریک‌ها (آخرین ۲۰ مقدار)</div>
 <div class="card-body p-0">
 <div class="table-responsive">
 <table class="table table-sm mb-0">
```

```

<thead class="table-light"><tr><th>زمان</th><th>CPU%</th><th>RAM%</th><th>Temp</th><th>IP</th><th>hog</th></tr></thead>

<tbody id="metricsT"></tbody>

</table>
</div>
</div>
</div>
</div>
</div>

<div class="col-lg-6">
<div class="card shadow-sm">
<div class="card-header">خلاصه وضعیت</div>
<div class="card-body">
<ul class="list-group list-group-flush small">
<li class="list-group-item d-flex justify-content-between">وضعیت:—
<li class="list-group-item d-flex justify-content-between">IP:—

</div>
</div>
</div>
</div>
</div>

<script src="assets/app.js"></script>
<script> IOT.initMetricsPage(); </script>

```

توضیح :

## ۱ هدر مشترک و استایل‌ها

```
<?php include '_top.php'; ?>
```

- اسکلت/ناویار/بررسی لاین را از فایل مشترک `_top.php` شروع می‌آورد. ... آنجاست

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="assets/style.css">
```

  - برای گرید، کارت، جدول و... Bootstrap CSS
  - برای آیکن‌ها (اگر جایی استفاده شود). Bootstrap Icons
  - استایل اختصاصی پروژه از `assets/style.css`.

کانتینر و گرید

```
<div class="container py-4">
```

```
<div class="row g-3">
```

- عرض واکنش‌گرا با فاصله عمودی. container: (py-4)
- یک ردیف شبکه با فاصله (gap) بین ستون‌ها. row g-3:

«کارت نمونه متريک‌ها»

```
<div class="col-lg-6">
```

```
<div class="card shadow-sm">
```

```
<div class="card-header">نمونه متريک‌ها (آخرین ۲۰ مقدار)</div>
```

- ستون در نمایشگر بزرگ نصف عرض. (col-lg-6)

کارت با سایه‌ی کم •

هدر کارت: عنوان جدول متريکها. •

```
<div class="card-body p-0">
 <div class="table-responsive">
 <table class="table table-sm mb-0">
 <thead class="table-light">

<tr><th>زمان</th><th>CPU%</th><th>RAM%</th><th>Temp</th><th>IP</th><th>hog</th>
</tr>
 </thead>
 <tbody id="metricsT"></tbody>
 </table>
 </div>
</div>
```

card-body p-0: محتوای کارت بدون پدینگ تا جدول تمام عرض باشد. •

table-responsive: اگر جدول پهن شد، اسکرول افقی اضافه می‌کند. •

table table-sm mb-0: جدول فشرده با فاصله‌ی پایانی صفر. •

thead table-light: هدر روشِ جدول. •

ستون‌ها: «زمان، IP، Temp، RAM%， CPU%». •

tbody id="metricsT": جدول که با جاوا اسکریپت پر می‌شود (آخرین ۲۰ رکورد). •

بخش «hog» معمولاً برای نمایش پُرخورترین پردازه/منبع است مثلاً اسم پروسه‌ای که بیشترین CPU/RAM را گرفته — هرچی اسکریپت شما در () IOT.initMetricsPage() تنظیم کند.

4ستون راست: کارت «خلاصه وضعیت»

```
<div class="col-lg-6">
```

```

<div class="card shadow-sm">
 <div class="card-header">خلاصه وضعیت</div>
 <div class="card-body">
 <ul class="list-group list-group-flush small">
 <li class="list-group-item d-flex justify-content-between">
 وضعیت—

 <li class="list-group-item d-flex justify-content-between">
 IP:—

 </div>
</div>

```

- کارت دوم برای نمایش خلاصه:
- ردیف «وضعیت» «متلأ → Online/Offline/Degraded) #گذاشته می‌شود.
- ردیف «IP» «دستگاه/نود → مقدار در #ip #گذاشته می‌شود.
- list-group-flush: اضافه داخل کارت.
- small: کمی کوچک‌تر.
- justify-content-between: عناوan سمت راست، مقدار سمت چپ در RTL معکوس بصری

## 15 اسکریپت‌ها و راهاندازی صفحه

```

<script src="assets/app.js"></script>
<script> IOT.initMetricsPage(); </script>

```

فایل جاوا اسکریپت اصلی پروژه. انتظار می‌رود آبجکت سراسری IOT را تعریف کرده باشد.

• فراخوانی IOT.initMetricsPage() برای:

○ کردن داده‌های متریک (مثلاً از api/system\_health.php

.api/metrics\_last20.php)

○ پرکردن T metrics با ۲۰ ردیف آخر،

○ مقداردهی st #ip و

اگر در کنسول خطای «IOT is not defined» دیدی، یعنی assets/app.js قبل از این صفحه لود نشده یا IOT.initMetricsPage داخلش تعریف نشده.

## نکات کاربردی و پیشنهادی

• پر کردن جدول در JS: ساخت هر ردیف

// نمونه‌ی ساده‌ی رندر هر ردیف

```
• function rowHtml(m) {
 • // m: {ts,cpu_percent,ram_percent,cpu_temp_c,ip,top_proc}
 • return `<tr>
 • <td>${new Date(m.ts).toLocaleString('fa-IR')}</td>
 • <td>${m.cpu_percent?.toFixed?.(1) ?? '—'}</td>
 • <td>${m.ram_percent?.toFixed?.(1) ?? '—'}</td>
 • <td>${m.cpu_temp_c?.toFixed?.(1) ?? '—'}°C</td>
 • <td>${m.ip ?? '—'}</td>
 • <td>${m.top_proc ?? '—'}</td>
 • </tr>`;
 • }
```

• هایلایت وضعیت: می‌توانی badge را با st # را با رنگی ست کنی (مثلاً سبز/قرمز):

const st = document.getElementById('st');

- st.textContent = statusText; // 'Online' | 'Offline' •
- st.className = 'badge ' + (statusOk ? 'bg-success' : 'bg-danger'); •
- واحد دما: اگر سنسور گاهی null برگرداند، حتماً «—» نمایش بده تا جدول نشکند. •
- بهینه‌سازی رندر: قبل از افزودن ۲۰ ردیف به DOM، از DocumentFragment یا innerHTML یکجا استفاده کن تا ریفلو کم شود. •
- محل API اطمینان از وجود CORS می‌کند مسیر درست، endpoints() که IOT.initMetricsPage() مصرف می‌کند JSON صحیح •

#### processes.php:

```
<?php include '_top.php'; ?>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css">

<div class="container py-4">

<div class="card shadow-sm">
 <div class="card-body p-0">
 <div class="table-responsive">
 <table class="table table-hover table-sm mb-0">
 <thead class="table-light">
 <tr>
 <th>#</th><th>PID</th><th>کاربر</th><th>نام</th><th>CPU%</th><th>RAM%</th><t
h>رمان</th><th>عملیات</th>
 </tr>
 <tbody>
 <tr>
 <td>1</td>
 <td>12345678901234567890</td>
 <td>user1</td>
 <td>John Doe</td>
 <td>50</td>
 <td>10</td>
 <td>Windows 10 Pro</td>
 <td>
 مشاهده ویرایش حذف
 </td>
 </tr>
 </tbody>
 </table>
 </div>
 </div>
</div>
```

```

</thead>

<tbody id="procFull"></tbody>

</table>

</div>

</div>

</div>

</div>

<script src="assets/app.js"></script>

<script> IOT.initProcessesPage(); </script>

```

توضیح :

?>php include '\_top.php'<? ;'

فایل مشترک بالاسری را لود می کند (احراز هویت/هدر/منو/متا - بسته به پیاده سازی شما).

>link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"

rel="stylesheet"<"

>link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet"<"

>link rel="stylesheet" href="assets/style.css"<"

استایل های بوت استرپ، آیکون ها و CSS اختصاصی شما را اضافه می کند.

>div class="container py-4"<"

ظرف اصلی صفحه با فاصله عمودی (padding بالا/پایین).

> div class="card shadow-sm"<"

```
> div class="card-body p-0<"
```

```
> div class="table-responsive<"
```

کارت با سایه ملایم؛ بدن کارت بدون padding و یک ظرف «table-responsive» تا جدول در موبایل اسکرول افقی بگیرد.

```
> table class="table table-hover table-sm mb-0<"
```

```
> thead class="table-light<"
```

```
> tr<
```

```
> th>#</th><th>PID</th><th>کاربر</th><th>نام</th><th>CPU%</th><th>RAM%</th><th>عملیات</th><th>زمان</th>
```

```
/> tr<
```

```
/> thead<
```

جدول کم حجم (table-sm) با افکت hover و بدون مارجین پایین.

هدر روشن، ستون‌ها:

#: شماره ردیف در همین صفحه.

PID: شناسه پردازه.

کاربر: صاحب پردازه (username).

نام: نام پردازه/فرآیند.

.CPU: درصد مصرف CPU

RAM: درصد مصرف حافظه.

زمان: زمان اجرا/uptime یا start time (بسته به داده‌ای که می‌فرستید).

عملیات: دکمه‌های کشی (متلاً Kill/Inspect) که بعداً با JS درج می‌شوند.

```
> tbody id="procFull"></tbody<
```

بدنه جدول که به صورت داینامیک توسط جاوااسکریپت پر می‌شود (ردیف‌های پردازه‌ها).

```
/> div> <!-- .table-responsive<--
```

```
/> div> <!-- .card-body<--
```

```
/> div> <!-- .card<--
```

```
/>div> <!-- .container<--
```

بستن تگ‌ها؛ ساختار کارت داخل کانتینر.

```
>script src="assets/app.js"></script<
```

```
>script> IOT.initProcessesPage(); </script<
```

فایل JS اصلی پروژه را لود می‌کند.

IOT.initProcessesPage() را اجرا می‌کند تا:

داده‌های پردازه‌ها را از API/endpoint شما واکشی کند،

#procFull را با ردیف‌ها پر کند.

ایونت‌هندلرها را برای صفحه‌بندی/مرتب‌سازی/اکشن‌های ستون «عملیات» متصل کند (اگر پیاده‌سازی شده باشد).

### System.PHP:

```
<?php include '_top.php'; ?>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.css" rel="stylesheet">

<link rel="stylesheet" href="assets/style.css">

<div class="container py-4">
 <div class="row g-3">
 <!-- مشخصات پایه -->
 <div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header">
 <i class="bi bi-pc-display me-1"></i> مشخصات سیستم
 </div>
 <div class="card-body">
 <table class="table table-sm mb-0">
 <tbody id="specsT">
 <tr><th>مدل دستگاه</th><td id="dev">—</td></tr>
 <tr><th>Hostname</th><td id="host">—</td></tr>
 <tr><th>سیستم عامل</th><td id="os">—</td></tr>
 <tr><th>کرنل</th><td id="kernel">—</td></tr>
```

```

<tr><th>معماری</th><td id="arch">—</td></tr>
<tr><th>CPU</th><td id="cpu">—</td></tr>
<tr><th>تعداد هسته</th><td id="cores">—</td></tr>
</tbody>
</table>
</div>
</div>
</div>


```

```

<!-- KPI لحظه‌ای -->
<div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header">
 <i class="bi bi-speedometer2 me-1"></i> وضعیت لحظه‌ای
 </div>
 <div class="card-body">
 <div class="row g-3">
 <div class="col-4">
 <div class="kpi">
 <div class="card-body">
 <div class="kpi-label">CPU</div>
 <div class="kpi-value" id="cpuK">—</div>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
</div>

```

```
<div class="col-4">
 <div class="kpi">
 <div class="card-body">
 <div class="kpi-label">RAM</div>
 <div class="kpi-value" id="ramK">—</div>
 </div>
 </div>
 </div>
 <div class="col-4">
 <div class="kpi">
 <div class="card-body">
 <div class="kpi-label">Temp</div>
 <div class="kpi-value" id="tmpK">—</div>
 </div>
 </div>
 </div>
 </div>
 <div class="mt-3 small">
 IP: —
 RSSI: —
 </div>
 </div>
 </div>
<!-- RAM -->
```

```

<div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header"><i class="bi bi-memory me-1"></i> حافظه (RAM)</div>
 <div class="card-body">
 <table class="table table-sm mb-3">
 <tbody>
 <tr><th>کل</th><td id="ram_total">—</td></tr>
 <tr><th>مصرف شده</th><td id="ram_used">—</td></tr>
 <tr><th>برابر</th><td id="ram_free">—</td></tr>
 </tbody>
 </table>
 <div class="progress" style="height: 10px;">
 <div id="ram_bar" class="progress-bar" role="progressbar"
 style="width:0%">0%</div>
 </div>
 </div>
 </div>
</div>

```

```

<!-- Disk / SD -->
<div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header"><i class="bi bi-sd-card me-1"></i> فضای ذخیره سازی
 (کارت / دیسک)</div>
 <div class="card-body">
 <table class="table table-sm mb-3">

```

```

<tbody>
<tr><th>کل</th><td id="disk_total">—</td></tr>
<tr><th>صرف شده</th><td id="disk_used">—</td></tr>
<tr><th>آزاد</th><td id="disk_free">—</td></tr>
</tbody>
</table>

<div class="progress" style="height: 10px;">
 <div id="disk_bar" class="progress-bar bg-info" role="progressbar"
 style="width:0%">0%</div>
</div>
</div>
</div>
</div>

</div>
</div>
<!-- Power Controls -->
<!-- Power Controls -->
<div class="col-12 mt-3">
 <div class="card shadow-sm card-glass">
 <div class="card-header">
 <i class="bi bi-power me-1"></i> کنترل قدرت (سیستم)
 </div>
 <div class="card-body d-flex gap-2 flex-wrap">

```

```

<!-- دکمه ریاستارت (مشکی) -->
<button class="btn btn-reboot"
 onclick="IOT.sysPower('reboot')">
 <i class="bi bi-arrow-repeat me-1"></i> ری استارت
</button>

<!-- دکمه خاموش (مشکی) -->
<button class="btn btn-shutdown"
 onclick="IOT.sysPower('shutdown')">
 <i class="bi bi-power me-1"></i> خاموش
</button>

<div class="text-muted small ms-auto">
 <i class="bi bi-info-circle me-1"></i>
 پس از اجرای دستور، ارتباط وب قطع می شود (طبیعی است).
</div>
</div>
</div>

<script src="assets/app.js?v=6"></script>
<script>IOT.initSystemPage();</script>
<link rel="stylesheet" href="assets/style.css?v=8">

```

توضیح:

## ابتدای صفحه

```
<?php include '_top.php'; ?>
```

بارگذاری می‌شود (معمولاً شامل سشن و هدر و منو) `_top.php` فایل مشترک.

```
<link href="...bootstrap.min.css" rel="stylesheet">
```

```
<link href="...bootstrap-icons.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="assets/style.css">
```

بارگذاری استایل بوت‌استرپ، آیکون‌ها و استایل اختصاصی.

## ظرف اصلی

```
<div class="container py-4">
```

```
<div class="row g-3">
```

یک کانتینر با فاصله عمودی. درون آن یک ردیف با فاصله ۳ بین ستون‌ها.

## کارت «مشخصات سیستم»

```
<div class="col-lg-6">
```

```
<div class="card shadow-sm">
```

```
<div class="card-header">
```

```
<i class="bi bi-pc-display me-1"></i> مشخصات سیستم
```

```
</div>
```

```
<div class="card-body">
```

```
<table class="table table-sm mb-0">
```

```
<tbody id="specsT">
```

ستون نصف عرض. •

کارت با عنوان «مشخصات سیستم».

جدول کوچک با بدنه‌ای که بعداً با جوا اسکریپت پر می‌شود.

ردیفها:

مدل دستگاه (id="dev")

Hostname (id="host")

سیستم‌عامل (id="os")

کرنل (id="kernel")

معماری (id="arch")

CPU (id="cpu")

تعداد هسته‌ها (id="cores")

کارت «وضعیت لحظه‌ای»

```
<div class="col-lg-6">
 <div class="card shadow-sm">
 <div class="card-header">
 <i class="bi bi-speedometer2 me-1"></i> وضعیت لحظه‌ای
 </div>
```

ستون دوم نصف عرض.

کارت با عنوان «وضعیت لحظه‌ای».

داخل کارت:

یک ردیف با سه ستون هر کدام یک KPI

→ مقدار در CPU id="cpuK" ○

→ مقدار در RAM id="ramK" ○

`id="tmpK"` مقدار در `Temp` → ○

زیر **KPI** ها دو مقدار متنی: •

IP (`id="ipK"`) ○

RSSI (`id="rssik"`) ○

### کارت «حافظه» (RAM)

<div class="col-lg-6">

<div class="card shadow-sm">

<div class="card-header"><i class="bi bi-memory me-1"></i> حافظه (RAM)</div>

ستون نصف عرض. •

کارت با عنوان RAM •

داخل کارت:

جدول با سه ردیف: کل (`id="ram_total"`) ، مصرف شده (`id="ram_used"`) ، آزاد. (•

نوار پیشرفت (`id="ram_bar"`) برای نمایش درصد مصرف. •

### کارت «فضای ذخیره سازی» (Disk/SD)

<div class="col-lg-6">

<div class="card shadow-sm">

<div class="card-header"><i class="bi bi-sd-card me-1"></i> فضای ذخیره سازی

(کارت/دیسک)</div>

ستون دوم نصف عرض. •

کارت با عنوان ذخیره سازی. •

داخل کارت:

جدول با سه ردیف: کل (`id="disk_total"`) ، مصرف شده (`id="disk_used"`) ، آزاد. (•

- نوار پیشرفت (id="disk\_bar") برای درصد استفاده از دیسک.

## کارت «کنترل قدرت»

```
<div class="col-12 mt-3">
<div class="card shadow-sm card-glass">
<div class="card-header">
<i class="bi bi-power me-1"></i> کنترل قدرت (سیستم)
</div>
<div class="card-body">
 سطون تمام عرض.
</div>
<div class="card-footer">
 کارت با عنوان «کنترل قدرت».
</div>
<div class="card-glass">
 داخل کارت:
 دکمه ریاستارت:
 دکمه خاموش:
 متن هشدار: بعد از اجرا ارتباط وب قطع می‌شود (طبیعی است).
</div>

```

## اسکریپت‌ها

```
<script src="assets/app.js?v=6"></script>
<script>IOT.initSystemPage();</script>
<link rel="stylesheet" href="assets/style.css?v=8">
```

- فایل جاواسکریپت پروژه بارگذاری می‌شود.
- تابع () IOT.initSystemPage() اجرا می‌شود تا داده‌ها را واکشی و بخش‌های صفحه را پر کند.

- استایل نسخه‌ی جدید دوباره بارگذاری می‌شود.

## خلاصه عملکرد

این صفحه داشبوردی است برای:

- نمایش مشخصات پایه دستگاه مدل، سیستم‌عامل، کرنل، معماری، CPU
- نمایش وضعیت لحظه‌ای (CPU، IP، RAM، دما، RSSI).
- نشان دادن جزئیات RAM و Disk همراه با نوار پیشرفت.
- ارائه کنترل‌های سیستم (ری‌استارت و خاموش).
- همه داده‌ها توسط IOT.initSystemPage() به روزرسانی می‌شوند.

### API

#### \_db.php:

```
<?php

// api/_db.php — build a single PDO instance, reading config from ../../env.php

declare(strict_types=1);

// ---- Defaults (fallback) ----

$cfg = [
 'DB_HOST' => '127.0.0.1',
 'DB_NAME' => 'iot',
 'DB_USER' => 'iotuser',
 'DB_PASS' => 'iotpass',
 'DB_CHARSET' => 'utf8mb4',
];
```

```

$envPath = __DIR__ . '/../.env.php';

if (file_exists($envPath)) {
 $ret = include $envPath;

 // Case A: .env.php returns an array
 if (is_array($ret)) {
 $cfg = array_merge($cfg, array_intersect_key($ret, $cfg));
 }
}

// Case B: .env.php sets variables $DB_HOST, ... (not returned)
foreach(['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {
 if (isset($GLOBALS[$k]) && is_string($GLOBALS[$k]) && $GLOBALS[$k] !== "") {
 $cfg[$k] = $GLOBALS[$k];
 }
}

// Case C: .env.php defines constants define('DB_HOST', ...)
foreach(['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {
 if (defined($k) && constant($k) !== "") {
 $cfg[$k] = constant($k);
 }
}

// Case D (optional): environment variables
foreach(['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {

```

```

$v = getenv($k);

if ($v !== false && $v !== "") $cfg[$k] = $v;

}

}

// ---- Single PDO ----

function db(): PDO {
 static $pdo = null;

 if ($pdo instanceof PDO) return $pdo;

 $c = $GLOBALS['cfg'];

 $dsn =
"mysql:host={$c['DB_HOST']};dbname={$c['DB_NAME']};charset={$c['DB_CHARSET']}";

 $opts = [
 PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
 PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
 PDO::ATTR_EMULATE_PREPARES => false, // LIMIT/OFFSET
];

 $pdo = new PDO($dsn, $c['DB_USER'], $c['DB_PASS'], $opts);

 return $pdo;
}

```

توضیح :

هدر فایل

```
// api/_db.php — build a single PDO instance, reading config from ../../.env.php
declare(strict_types=1);
```

- توضیح: این فایل قرار است فقط یک اتصال PDO بسازد.

```
declare(strict_types=1); → PHP
 خودکار int نشود
```

### مقادیر پیشفرض

```
$cfg = [
 'DB_HOST' => '127.0.0.1',
 'DB_NAME' => 'iot',
 'DB_USER' => 'iotuser',
 'DB_PASS' => 'iotpass',
 'DB_CHARSET' => 'utf8mb4',
];
```

- یک آرایه‌ی پیشفرض برای اتصال: هاست، نام دیتابیس، کاربر، پسورد، و charset.
- اگر فایل کانفیگ پیدا نشد یا ناقص بود، این مقادیر fallback هستند.

### مسیر فایل env

```
$envPath = __DIR__ . '/../../.env.php';
if (file_exists($envPath)) {
 $ret = include $envPath;
```

- مسیر فایل env.php. یک پوشه بالاتر از /api ساخته می‌شود.
- اگر فایل وجود دارد، include می‌شود و نتیجه‌اش در \$ret ذخیره می‌شود.

### Case A. آرایه برگرداند env.php اگر

```

if (is_array($ret)) {

 $cfg = array_merge($cfg, array_intersect_key($ret, $cfg));

}

```

- اگر خروجی فایل یک آرایه باشد، مقادیر مشترک کلیدهایش روی \$cfg merge می‌شود.
- یعنی فقط کلیدهای معتبر (...DB\_HOST) وارد می‌شوند.

#### اگر متغیرهای سراسری ساخته شده باشند Case B:

```

foreach(['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {

 if (isset($GLOBALS[$k]) && is_string($GLOBALS[$k]) && $GLOBALS[$k] !== "") {

 $cfg[$k] = $GLOBALS[$k];

 }
}

```

- اگر فایل .env.php معرفی‌هایی مثل \$DB\_HOST نه آرایه return، اینجا مقدارشان وارد \$cfg می‌شود.

#### اگر ثابت (constant) تعریف شده باشد Case C

```

foreach(['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {

 if (defined($k) && constant($k) !== "") {

 $cfg[$k] = constant($k);

 }
}

```

- اگر فایل .env.php چیزی مثل define('DB\_HOST','localhost') داشته باشد، اینجا خوانده و در \$cfg قرار داده می‌شود.

#### اگر در Environment Variable باشند Case D

```

foreach (['DB_HOST','DB_NAME','DB_USER','DB_PASS','DB_CHARSET'] as $k) {
 $v = getenv($k);
 if ($v !== false && $v !== "") $cfg[$k] = $v;
}

```

- در آخر هم، اگر متغیرهای محیطی سیستم مثلاً در Docker یا سرور تنظیم شده باشند، روی `$cfg override` می‌شوند.

## تابع(db())

```

function db(): PDO {
 static $pdo = null;
 if ($pdo instanceof PDO) return $pdo;

```

- تابع db() یک Singleton است:

- بار اول که صدا زده شود، PDO می‌سازد.
- دفعه‌های بعد همان شیء PDO قبلی را برمی‌گرداند.

## DSN ساخت

```

$c = $GLOBALS['cfg'];
$dsn =
"mysql:host={$c['DB_HOST']};dbname={$c['DB_NAME']};charset={$c['DB_CHARSET']}";

```

- مقادیر از `$cfg` گرفته می‌شود.

- آدرس MySQL اتصال DSN ساخته می‌شود.

## گزینه‌های PDO

```
$opts = [
```

```

PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
PDO::ATTR_EMULATE_PREPARES => false, // LIMIT/OFFSET
]؛
 • هر خطا به شکل Exception پرتاب شود.
 • کوئری‌ها به صورت آرایه‌ی associative برگردند (نه عددی).
 • استفاده MySQL از PDO باعث می‌شود EMULATE_PREPARES=false:
 • کند (بهتر برای امنیت و بهینه‌سازی). همچنین وقتی LIMIT یا OFFSET عددی می‌فرستیم، درست کار می‌کند.

```

## ساخت اتصال و بازگشت

```

$pdo = new PDO($dsn, $c['DB_USER'], $c['DB_PASS'], $opts);
return $pdo;
}

```

- اتصال PDO ساخته می‌شود و در متغیر static ذخیره می‌شود.
- برگردانده می‌شود.

## خلاصه عملکرد

این فایل یک اتصال واحد PDO به دیتابیس MySQL می‌سازد.

ترتیب خواندن تنظیمات:

1. مقادیر پیش‌فرض داخلی.
2. آرایه‌ی بازگشتی از .env.php.
3. متغیرهای سراسری DB\_HOST
4. ثوابت تعریف شده. (define('DB\_HOST',...)).
5. متغیرهای محیطی سیستم (env).

سپس تابع db() ساخته می شود که یکبار اتصال برقرار می کند و در دفعات بعدی همان اتصال را باز می گرداند.

## API

### Events\_list.php:

```
<?php

// api/events_list.php

declare(strict_types=1);

header('Content-Type: application/json; charset=utf-8');

require_once __DIR__ . '/_db.php';

$action = $_GET['action'] ?? "";

$q = isset($_GET['q']) ? trim((string)$_GET['q']) : "";
$start = isset($_GET['start']) ? str_replace('T', '', trim((string)$_GET['start'])) : "";
$end = isset($_GET['end']) ? str_replace('T', '', trim((string)$_GET['end'])) : "";
$confMin = isset($_GET['conf_min']) ? trim((string)$_GET['conf_min']) : "";
$confMax = isset($_GET['conf_max']) ? trim((string)$_GET['conf_max']) : "";

$page = isset($_GET['page']) ? max(1, intval($_GET['page'])) : 1;
$per = isset($_GET['per']) ? max(1, min(100, intval($_GET['per']))) : 50;
$HARD_CAP = 1000;

$where = " WHERE 1=1 ";
$bind = [];
```

```

/**
 * استفاده نشود تا ایندکس‌ها قابل استفاده باشند (نکته LOWER(...)).
 * (غیرحساس به حروف) برای ستون‌های متنی است ci_‌های collation فرض بر
 */
if ($q !== "") {
 // تغییر دهید "$q%" بهتر عمل کند، الگو را به آن اگر می‌خواهید ایندکس.
 $where .= " AND (label LIKE :q1 OR node_id LIKE :q2)";
 $bind[':q1'] = "%$q%";
 $bind[':q2'] = "%$q%";
}

if ($start !== "") { $where .= " AND ts >= :s"; $bind[':s'] = $start; }
if ($end !== "") { $where .= " AND ts <= :e"; $bind[':e'] = $end; }

if ($confMin !== "") {
 $cm = max(0.0, min(100.0, floatval($confMin)));
 $where .= " AND confidence >= :cmin ";
 $bind[':cmin'] = $cm;
}

if ($confMax !== "") {
 $cx = max(0.0, min(100.0, floatval($confMax)));
 $where .= " AND confidence <= :cmax ";
 $bind[':cmax'] = $cx;
}

```

```

try{
 $pdo = db();

 // ساخت خودکار ایندکس‌ها -- --
 if ($action === 'index') {
 // در MySQL 8+ دسترس است IF NOT EXISTS در
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts ON audio_events
(ts)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_nodeid ON audio_events
(node_id)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_label ON audio_events
(label)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_conf ON audio_events
(confidence)");

 // در صورت نیاز به فیلتر همزمان زمان+کانفیدنس :
 // $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts_conf ON audio_events
(ts, confidence)");
 }

 echo json_encode(['ok'=>true, 'message'=>'Indexes ensured (ts, node_id, label,
confidence).'], JSON_UNESCAPED_UNICODE);

 exit;
}

// شمارش برای صفحه‌بندی --
$sqlCount = "SELECT COUNT(*) FROM audio_events $where";
$cntSt = $pdo->prepare($sqlCount);
foreach($bind as $k=>$v) {

```

```

$cntSt->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_STR);

}

$cntSt->execute();

$total = (int)$cntSt->fetchColumn();

$total_capped = min($total, $HARD_CAP);

$pages = max(1, (int)ceil($total_capped / $per));

if ($page > $pages) $page = $pages;

$off = ($page - 1) * $per;

// -- کوئری داده‌ها

$sqlData = "SELECT id, ts, label, confidence, node_id
 FROM audio_events
 $where
 ORDER BY ts DESC
 LIMIT :lim OFFSET :off";

$st = $pdo->prepare($sqlData);

foreach($bind as $k=>$v) {

 $st->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_STR);

}

$st->bindValue(':lim', $per, PDO::PARAM_INT);

$st->bindValue(':off', $off, PDO::PARAM_INT);

// -- ساده کار مجاز (اجرا زمان کل)


```

```

$t0 = microtime(true);

$st->execute();

$rows = $st->fetchAll();

$elapsed_ms = (microtime(true) - $t0) * 1000.0;

// اگر MySQL 8+: EXPLAIN ANALYZE درخواستی: پلن اجرا -- EXPLAIN
// و گرنج EXPLAIN

$explain = null;

if ($action === 'bench') {

 $sqlExplain = "EXPLAIN ANALYZE $sqlData";

 try {

 $est = $pdo->prepare($sqlExplain);

 foreach($bind as $k=>$v) {

 $est->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_STR);

 }

 $est->bindValue(':lim', $per, PDO::PARAM_INT);

 $est->bindValue(':off', $off, PDO::PARAM_INT);

 $est->execute();

 $explain = $est->fetchAll();

 } catch (\Throwable $e) {

 $sqlExplain = "EXPLAIN $sqlData";

 $est = $pdo->prepare($sqlExplain);

 foreach($bind as $k=>$v) {

 $est->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_STR);

 }

 $est->bindValue(':lim', $per, PDO::PARAM_INT);

 $est->bindValue(':off', $off, PDO::PARAM_INT);

 }

}


```

```

$est->execute();

$explain = $est->fetchAll();

}

}

echo json_encode([
'rows' => $rows,
'page' => $page,
'per' => $per,
'total' => $total_capped,
'pages' => $pages,
'bench_ms' => round($elapsed_ms, 2),
'explain' => $explain
], JSON_UNESCAPED_UNICODE);

} catch (\Throwable $ex) {
 http_response_code(500);
 echo json_encode(['ok'=>false,'error'=>'db_error','message'=>$ex->getMessage()]);
 JSON_UNESCAPED_UNICODE);
}

```

توضیح:

### هدر و پیکربندی خروجی

```

// api/events_list.php

declare(strict_types=1);

header('Content-Type: application/json; charset=utf-8');

```

توضیح فایل.

• برسی سختگیرانه‌ی تایپ‌ها. `strict_types=1`

• هدر پاسخ JSON با یونیکد.

## بارگذاری اتصال (DB (PDO singleton))

```
require_once __DIR__ . '/_db.php';
```

• فایل `_db.php` را می‌آورد تا تابع `db()` یک PDO مشترک بسازد.

## پارامترهای ورودی

```
$action = $_GET['action'] ?? '';
```

```
$q = isset($_GET['q']) ? trim((string)$_GET['q']) : '';
```

```
$start = isset($_GET['start']) ? str_replace('T', trim((string)$_GET['start'])) : '';
```

```
$end = isset($_GET['end']) ? str_replace('T', trim((string)$_GET['end'])) : '';
```

```
$confMin = isset($_GET['conf_min']) ? trim((string)$_GET['conf_min']) : '';
```

```
$confMax = isset($_GET['conf_max']) ? trim((string)$_GET['conf_max']) : '';
```

• `action` حالت‌های ویژه مثل ساخت ایندکس یا بنچمارک.

• فیلترها:

○ `label/node_id`: جستجوی متنی روی `q`.

○ `start/end`: بازه‌ی زمانی، `T` را به فاصله تبدیل می‌کند (فرمت `YYYY-MM-DD HH:MM`).

○ `conf_min/conf_max`: بازه‌ی `confidence` (`0-100`).

```
$page = isset($_GET['page']) ? max(1, intval($_GET['page'])) : 1;
```

```
$per = isset($_GET['per']) ? max(1, min(100, intval($_GET['per']))) : 50;
```

```
$HARD_CAP = 1000;
```

• صفحه‌بندی: صفحه  $\leq 1$ ، اندازه‌ی صفحه بین ۱ تا ۱۰۰ (پیش‌فرض ۵۰).

• سقف سفت و سخت (HARD\_CAP) برای مجموع ردیفهای قابل گزارش جلوگیری از پاسخهای خیلی بزرگ

## مونتاژ WHERE و بایندها

```
$where = " WHERE 1=1 ";
```

```
$bind = [];
```

• اسکلت WHERE برای افزودن شرطها.

```
/**
```

\* استفاده نشود تا ایندکس‌ها قابل استفاده باشند (نکته LOWER(...)).

\* (غیرحساس به حروف) برای ستون‌های متنی است ci\_\* های collation فرض بر.

```
*/
```

```
if ($q !== "") {
```

// تغییر دهید "\$q%" بهتر عمل کند، الگو را به label/node\_id اگر می‌خواهید ایندکس.

```
$where .= " AND (label LIKE :q1 OR node_id LIKE :q2) ";
```

```
$bind[':q1'] = "%$q%";
```

```
$bind[':q2'] = "%$q%";
```

```
}
```

• خیلی مهم: از LOWER() روی ستون‌ها استفاده نشده تا ایندکس از کار نیافتد. فرض بر ci\_\* های collation تا جست‌وجوی غیرحساس به حروف بدون LOWER ممکن شود.

• نوک‌و-دُم‌دار است و ایندکس را خنثی می‌کند. اگر ممکن است به الگوی q% تغییرش بده تا

• ایندکس B-Tree روی label/node\_id قابل استفاده شود.

```
if ($start !== "") { $where .= " AND ts >= :s "; $bind[':s'] = $start; }
```

```
if ($end !== "") { $where .= " AND ts <= :e "; $bind[':e'] = $end; }
```

• فیلتر بازه‌ی زمانی روی ts ایندکس روی ts بسیار کارآمد است.

```
if ($confMin !== "") {
```

```
$cm = max(0.0, min(100.0, floatval($confMin)));
```

```
$where .= " AND confidence >= :cmin ";
```

```

$bind[':cmin'] = $cm;
}

if ($confMax !== '') {
 $cx = max(0.0, min(100.0, floatval($confMax)));
 $where .= " AND confidence <= :cmax ";
 $bind[':cmax'] = $cx;
}

```

• محدوده‌ی **confidence** با نگهبان‌های عددی (۰ تا ۱۰۰).

• ایندکس روی **confidence** کمک می‌کند؛ ولی اگر همزمان **ts** فیلتر می‌شود، ایندکس مرکب، **ts** می‌تواند بهتر باشد.

## اتصال و مسیرهای ویژه

```

$pdo = db();

PDO. گرفتن •

-- // ساخت خودکار ایندکس‌ها

if ($action === 'index') {
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts ON audio_events
(ts)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_nodeid ON audio_events
(node_id)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_label ON audio_events
(label)");
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_conf ON audio_events
(confidence)");
 // $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts_conf ON audio_events
(ts, confidence)");
}

```

```

echo json_encode(['ok'=>true, 'message'=>'Indexes ensured (ts, node_id, label,
confidence).'], JSON_UNESCAPED_UNICODE);

exit;

}

```

- حالت **action=index** ایندکس‌ها را تضمین می‌کند.
- ستون‌های ایندکس شده : ts, node\_id, label, confidence.
- (اختیاری) ایندکس مرکب ts, confidence برای کوئری‌های دوشرطی.

#### شمارش برای صفحه‌بندی

```

$sqlCount = "SELECT COUNT(*) FROM audio_events $where";

$cntSt = $pdo->prepare($sqlCount);

foreach($bind as $k=>$v) {

 $cntSt->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_INT);

}

$cntSt->execute();

$total = (int)$cntSt->fetchColumn();

```

- اجرای COUNT(\*) برای دانستن تعداد کل مطابق فیلتر.

نکته کوچک : در هر دو حالت PDO::PARAM\_STR بایند می‌کنی؛ برای دقیق‌تر، عددی‌ها را بده (برای تاریخ/متن همان PDO::PARAM\_INT خوب است)

```

$total_capped = min($total, $HARD_CAP);

$pages = max(1, (int)ceil($total_capped / $per));

if ($page > $pages) $page = $pages;

$off = ($page - 1) * $per;

```

- اعمال سقف خروجی، محاسبه تعداد صفحات، تصحیح صفحه‌ی فعلی، محاسبه‌ی offset.

## کوئری داده‌ها Select اصلی

```
$sqlData = "SELECT id, ts, label, confidence, node_id
FROM audio_events
$where
ORDER BY ts DESC
LIMIT :lim OFFSET :off";
```

- خروجی ستونی سبک.

ترتیب → ORDER BY ts DESC: با ایندکس روی ts این خیلی سریع می‌شود

- LIMIT/OFFSET. صفحه‌بندی با.

```
$st = $pdo->prepare($sqlData);
foreach($bind as $k=>$v) {
 $st->bindValue($k, $v, is_numeric($v) ? PDO::PARAM_STR : PDO::PARAM_INT);
}

$st->bindValue(':lim', $per, PDO::PARAM_INT);
$st->bindValue(':off', $off, PDO::PARAM_INT);
```

- پارامترها بایند می‌شوند. باز هم: می‌توانی برای INT باز: cmin/:cmax بسته به نوع ستون استفاده کنی;

:درست به INT بسته شده

```
$t0 = microtime(true);
$st->execute();
$rows = $st->fetchAll();
$elapsed_ms = (microtime(true) - $t0) * 1000.0;
```

- بنچمارک ساده: زمان اجرای SELECT در میلی‌ثانیه.

## پلن اجرا

```

$explain = null;

if ($action === 'bench') {

 $sqlExplain = "EXPLAIN ANALYZE $sqlData";

 try {

 $est = $pdo->prepare($sqlExplain);

 ...

 $explain = $est->fetchAll();

 } catch (\Throwable $e) {

 $sqlExplain = "EXPLAIN $sqlData";

 ...

 $explain = $est->fetchAll();

 }

}

```

اگر `action=bench` باشد:

- روی MySQL 8+: EXPLAIN ANALYZE پلن + زمان‌های واقعی اجرای هر مرحله.
- اگر خطا (قدیمی‌تر) EXPLAIN به `fallback` معمولی (پلن تخمینی).

## خروجی JSON و خطاپردازی

```

echo json_encode([
 'rows' => $rows,
 'page' => $page,
 'per' => $per,
 'total' => $total_capped,
 'pages' => $pages,
 'bench_ms' => round($elapsed_ms, 2),
]

```

```

'explain' => $explain
], JSON_UNESCAPED_UNICODE);

explain bench) در صورت SELECT + خروجی. داده‌ها + متادیتا + زمان اجرای) •

} catch (Throwable $ex) {
 http_response_code(500);
 echo json_encode(['ok'=>false,'error'=>'db_error','message'=>$ex->getMessage()]);
 JSON. پیام و ۵۰۰ کد: خطاهای کلی •

}

```

### خلاصه عملکرد و نکات ایندکس (خیلی مهم)

- ایندهکس‌ها: این API خودش با action=index ایندکس‌های کلیدی را می‌سازد.
- INDEX(ts) و مخصوصاً WHERE ts BETWEEN ... ORDER BY ts DESC LIMIT ... احیاتی برای پرفورمنس اصلی اینجاست.
- \*\_ci از LOWER() روى ستون‌ها اجتناب شده تا ایندکس قابل استفاده بماند، با LOWER: و COLLATION جستجو غیرحساس به حروف هست.
- صفحه‌بندی ORDER BY ts DESC LIMIT/OFFSET های با ایندکس ts خوب است؛ ولی برای OFFSET های خیلی بزرگ، روش seek یعنی نگهداشتن last\_ts/last\_id و استفاده از WHERE ts < :last بسیار سریع‌تر است.
- COUNT(\*) با همان WHERE اجرا می‌شود؛ اگر خیلی بزرگ شد، می‌توانی شمارش تقریبی یا cache شده داشته باشی.
- نوع بایند: برای INT باشد درست :off:lim:cmin:cmax: هم بهتر است نوع مناسب ستون PARAM\_STR (INT/DECIMAL) را بایند کنی؛ الان همه چیز به بسته می‌شود که کار می‌کند، ولی دقیق‌ترین نیست.

### ایندهکس روی دیتابیس (Database Index)

ایندکس در دیتابیس MySQL یک ساختار داده‌ای معمولاً **B-Tree** هست که روی ستون‌ها ساخته می‌شود.

- وقتی ایندکس داریم، موتور دیتابیس به جای اینکه جدول کامل را اسکن کند(Full Table Scan)، سریعاً به مکان رکورد موردنظر می‌رود.
- این دقیقاً مثل فهرست آخر کتاب است: برای پیدا کردن "موضوع X" لازم نیست همه صفحات را بخوانیم، مستقیم به شماره صفحه می‌پریم.

در کدی که نوشتیم، دستور ساخت ایندکس اینجاست:

```
if ($action === 'index') {
 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts ON audio_events
(ts)");

 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_nodeid ON audio_events
(node_id)");

 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_label ON audio_events
(label)");

 $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_conf ON audio_events
(confidence)");

 // ایندکس ترکیبی (زمان+اعتماد)
 // $pdo->exec("CREATE INDEX IF NOT EXISTS idx_audio_events_ts_conf ON audio_events
(ts, confidence)");
}
```

يعنى اين ستونها ايندکس مي شوند:

- زمان ثبت رويداد ts
- شناسه نود / دستگاه node\_id
- برچسب صوت label
- درصد اطمینان مدل confidence

ایندکس و کوئری (Query)

حالا ببینیم چرا این ایندکس‌ها مهم هستند و کجا استفاده می‌شوند:

### ۱ جستجو بر اساس زمان (ts)

کوئری:

```
SELECT ... FROM audio_events
WHERE ts >= :s AND ts <= :e
ORDER BY ts DESC
LIMIT 50 OFFSET 0;
```

- ایندکس روی **ts** اینجا خیلی مهم.
- بدون ایندکس → دیتابیس باید کل جدول را برای پیدا کردن ردیفها بررسی کند.
- با ایندکس → مستقیم به بازه زمانی پرش می‌کند (مثلاً از ساعت 10:00 تا 11:00) و سریعاً رکوردها را می‌خواند.
- همچنین برای MySQL می‌تواند از ایندکس به صورت معکوس استفاده کند و فقط ۵۰ ردیف آخر را برگرداند، نه کل جدول.

### ۲ جستجو بر اساس **node\_id** یا **label**

کوئری:

```
... WHERE label LIKE '%dog%' OR node_id LIKE '%sensor%'
```

- اینجا یک نکته خیلی مهم هست:
  - اگر جستجو با %متن (دوطرفه) باشد → ایندکس بی‌فایده است چون MySQL باید تمام مقادیر را چک کند.
  - اگر جستجو با متن % فقط اول کلمه باشد → ایندکس فعال می‌شود و کوئری سریع‌تر می‌شود.
- یعنی اگر بیشتر دنبال **prefix search** هستیم مثل "car%" ، ایندکس روی **label** و **node\_id** واقعاً به درد می‌خورد.

### ۳ جستجو بر اساس **confidence**

کوئری:

... WHERE confidence >= 70 AND confidence <= 90

- اگر ایندکس روی confidence داشته باشیم، دیتابیس سریع‌تر بازه‌ی اطمینان را پیدا می‌کند.
- اگر معمولاً همزمان روی ts و confidence فیلتر می‌کنیم، بهتر است یک ایندکس ترکیبی (ts, confidence) هم بسازیم.

#### ۴ شمارش ردیف‌ها

SELECT COUNT(\*) FROM audio\_events WHERE ...

- ایندهکس هم اینجا کمک می‌کند. چون اگر شرط روی ts یا confidence باشد، MySQL می‌تواند فقط ایندکس را بشمارد، نه تمام جدول را.

#### ابزار بررسی EXPLAIN و EXPLAIN ANALYZE

```
if ($action === 'bench') {
 $sqlExplain = "EXPLAIN ANALYZE $sqlData";
 ...
}
```

- با EXPLAIN می‌فهمیم کوئری چه پلنی دارد (مثلاً آیا از ایندکس استفاده می‌کند یا نه).
- با EXPLAIN ANALYZE در MySQL 8+ حتی زمان واقعی هر مرحله را نشان می‌دهد.
- ستون مهم در خروجی EXPLAIN:
  - key: ایندکسی که استفاده شده.
  - rows: تخمین تعداد ردیف‌هایی که باید اسکن شود.
  - key = NULL → اگر key = NULL یعنی ایندکس استفاده نشده، و کوئری احتمالاً کند است.

#### خلاصه عملکرد

- ایندهکس یعنی «میان‌بر» برای کوئری‌ها؛ بدونش جدول باید کامل اسکن شود.

- ایندکس  $\rightarrow$  حیاتی برای فیلتر/مرتب‌سازی زمانی.
- ایندکس  $\rightarrow$  LIKE 'abc%'prefix خوب است، اما فقط اگر جست‌جو به شکل 'باشد'label/node\_id.
- ایندکس  $\rightarrow$  مناسب برای **confidence** فیلتر محدوده اعتماد.
- ایندکس ترکیبی  $\rightarrow$  (ts, confidence) اگر زیاد این دو شرط را با هم استفاده می‌کنیم، عالی است.

## API

process\_kill.php:

```
<?php

// api/process_kill.php

session_start();

header('Content-Type: application/json; charset=utf-8');

if (!function_exists('str_starts_with')) {

 function str_starts_with($haystack, $needle){ return $needle!==" " &&
strpos($haystack,$needle)===0; }

}

function jexit($code, $arr){ http_response_code($code); echo json_encode($arr,
JSON_UNESCAPED_UNICODE); exit; }

function sh($cmd){

$des = [1=>'pipe','w'], 2=>'pipe','w'];

$p = proc_open($cmd, $des, $pipes);

$out=""; $err=""; $rc=null;

if(is_resource($p)){

 $out = stream_get_contents($pipes[1]); fclose($pipes[1]);

 $err = stream_get_contents($pipes[2]); fclose($pipes[2]);

}
```

```

$rc = proc_close($p);

}

return [$rc, $out, $err, $cmd];

}

function which_first($c){ foreach($c as $x){ if(@is_executable($x)) return $x; } return null; }

function web_user(){

if(function_exists('posix_geteuid') && function_exists('posix_getpwuid')){

$u = @posix_getpwuid(@posix_geteuid()); if(!empty($u['name'])) return $u['name'];

}

return getenv('APACHE_RUN_USER') ?: (getenv('USER') ?: 'www-data');

}

// --- Auth ---

if (empty($_SESSION['uid'])) jexit(401, ['ok'=>false,'error'=>'unauthorized']);

if ($_SERVER['REQUEST_METHOD']!='POST') jexit(405,
['ok'=>false,'error'=>'method_not_allowed']);

// --- Input ---

$pid = isset($_POST['pid']) ? intval($_POST['pid']) : 0;

if ($pid <= 1) jexit(400, ['ok'=>false, 'error'=>'bad_pid']);

// --- Paths ---

$BIN_PS = which_first(['/bin/ps','/usr/bin/ps']) ?: 'ps';

$BIN_KILL = which_first(['/bin/kill','/usr/bin/kill']);

$BIN_SUDO = which_first(['/usr/bin/sudo','/bin/sudo']);

if (!$BIN_KILL) jexit(500, ['ok'=>false,'error'=>'kill_not_found']);

```

```

// --- Proc info ---

list($_r1, $uOut, $uErr) = sh("$BIN_PS -o user= -p " . escapeshellarg($pid) . " 2>&1");
list($_r2, $cOut, $cErr) = sh("$BIN_PS -o comm= -p " . escapeshellarg($pid) . " 2>&1");
$user = trim($uOut); $comm = trim($cOut);
if ($user === "" || $comm === "") jexit(404,
['ok'=>false,'error'=>'not_found','stdout'=>$uOut,$cOut,'stderr'=>$uErr,$cErr]);

// --- Deny list ---

$deny = [
'systemd','init','kthreadd','rcu_','migration','ksoftirqd','watchdog','cpuhp','kworker','kdevtmpfs',
'sshd','apache2','httpd','nginx','php-fpm','php','mysqld','mariadb',
'dbus-daemon','systemd-udevd','systemd-journald','systemd-logind','systemd-resolved',
'NetworkManager','wpa_supplicant','rsyslogd','cron','crond','containerd','dockerd'
];
foreach($deny as $d){ if($comm === $d || str_starts_with($comm,$d)) jexit(403,
['ok'=>false,'error'=>'denied_critical','pid'=>$pid,'user'=>$user,'comm'=>$comm]); }

// --- Decide + Execute ---

$WEB = web_user();
$use_sudo = ($user !== $WEB); // اگر پروسیم مال وب سرور نیست، نیاز به sudo

// SIGTERM ، سپس در صورت لزوم SIGKILL
$prefix = $use_sudo && $BIN_SUDO ? "$BIN_SUDO -n " : "";
list($rc1,$out1,$err1,$cmd1) = sh($prefix.$BIN_KILL." -TERM ".escapeshellarg($pid)." 2>&1");
usleep(250000);

```

```

list($_rcA,$aliveOut1,$_errA) = sh("$BIN_PS -p ".escapeshellarg($pid)." -o pid= 2>&1");
$alive1 = trim($aliveOut1) != "";

if ($alive1) {

 list($rc2,$out2,$err2,$cmd2) = sh($prefix.$BIN_KILL." -KILL ".escapeshellarg($pid)." 2>&1");
 usleep(250000);

 list($_rcB,$aliveOut2,$_errB) = sh("$BIN_PS -p ".escapeshellarg($pid)." -o pid= 2>&1");
 $alive2 = trim($aliveOut2) != "";

 if ($alive2) {
 // نیاز به پسورد داشت، راهنمای روشن بده sudo اگر پیام
 if (strpos($err1.$err2.$out1.$out2, 'password is required') !== false) {
 jexit(403, [
 'ok'=>false,'error'=>'needs_sudoers',
 'message'=>'sudo NOPASSWD برای kill کن محدود است. مسیر درست /etc/sudoers.d/iotdash-kill ; visudo -cf /etc/sudoers.d/iotdash-kill',
 'hint'=>"echo 'www-data ALL=(root) NOPASSWD: /bin/kill, /usr/bin/kill' > /etc/sudoers.d/iotdash-kill ; visudo -cf /etc/sudoers.d/iotdash-kill",
 'stderr'=>$err1."\n".$err2, 'cmd_term'=>$cmd1, 'cmd_kill'=>$cmd2
]);
 }
 jexit(500, [
 'ok'=>false,'error'=>'kill_failed','pid'=>$pid,'user'=>$user,'comm'=>$comm,
 'message'=>'هنوز زنده است KILL و TERM پس از',
 'stderr'=>trim($err1."\n".$err2), 'stdout'=>trim($out1."\n".$out2)
]);
 }
}

```

```

 }
}

// OK

jexit(200, ['ok'=>true,'pid'=>$pid,'user'=>$user,'comm'=>$comm]);

```

توضیح :

```

?>php
//api/processes_list.php

header('Content-Type: application/json; charset=utf-8');

```

فایل API برای برگرداندن فهرست پردازه‌ها به صورت JSON

هدر پاسخ را روی application/json با یونیکد تنظیم می‌کند.

```
$limit = isset($_GET['limit']) ? max(1, min(100, intval($_GET['limit']))) : 20;
```

پارامتر limit از کوئری استرینگ خوانده می‌شود.

به عدد صحیح تبدیل، حداقل ۱ و حداکثر ۱۰۰؛ پیشفرض ۲۰.

```
*/
```

pid, user, comm, %cpu, %mem, lstart می‌گیریم:

```
/*
```

```
$cmd = "ps -eo pid,user,comm,%cpu,%mem,lstart --sort=-%cpu 2>/dev/null;"
```

دستور سیستم: ps با قالب ستون‌های مشخص:

شناسه پردازه pid

کاربر مالک user

نام فرمان (کوتاه) comm

CPU درصد مصرف cpu%.

RAM درصد مصرف mem%.

lstart زمان شروع پردازه (فرمت متنی طولانی)

.CPU sort=-%cpu-- یعنی مرتب‌سازی نزولی بر اساس CPU.

خطاهای استاندارد را دور می‌ریزد تا خروجی تمیز بماند. dev/null<2

```
$out = @shell_exec($cmd);
```

```
;[] = rows$
```

دستور اجرا می‌شود و خروجی کامل به صورت رشته در \$out می‌آید. @ خطاهای احتمالی را خاموش می‌کند.

آرایه‌ی نهایی ردیف‌ها را در \$rows جمع می‌کند.

```

if ($out)

$ lines = preg_split('/\r?\n/', trim($out));

if (count($lines) > 0) array_shift($lines); // header

```

اگر خروجی خالی نبود:

خروجی به خطوط جدا می‌شود.

ولین خط هدر ps است؛ حذف می‌شود.

```

$ i = 0;

foreach ($lines as $ln)

if ($i >= $limit) break;

```

شمارنده‌ی تعداد ردیف‌های اضافه شده.

اگر به سقف limit رسید، حلقه شکسته می‌شود.

```

// pid user comm cpu mem lstart...

if (preg_match('/^(\d+)\s+(\S+)\s+(\S+)\s+(\d\.\d+)\s+(\d\.\d+)\s+(.*$)', $ln, $m)) {

```

هر خط با یک عبارت منظم (Regex) پارس می‌شود:

$(\d+)$  → PID

$(\S+)$  → USER

(\S+) → COMM

([\d\.]\*) → %CPU

([\d\.]\*) → %MEM

→ بقیه‌ی خط = lstart به صورت متن (ممکن است فاصله داشته باشد) (\*.\*)

```
$ pid = (int)$m[1]
$ user = $m[2]
$ comm = $m[3]
$ cpu = floatval($m[4]);
$ mem = floatval($m[5]);
$ lstr = trim($m[6]);
```

نگاشت گروه‌های Regex به متغیرها و تبدیل‌های عددی لازم.

```
$ created_at = null;
$t = strtotime($lstr);
if ($t !== false) $created_at = date('Y-m-d H:i:s', $t);
```

تلاش برای تبدیل timestamp به lstart متنی به .strtotime

اگر موفق بود، به فرمت استاندارد Y-m-d H:i:s تبدیل می‌کند؛ و گرنه همان رشته‌ی اصلی بعداً استفاده می‌شود.

```
$ rows] = []
' pid' => $pid,
```

```

' user' => $user,
' name' => $comm,
' cpu_percent' => $cpu,
' mem_percent' => $mem,
' created_at' => $created_at ?: $lstr
;[
$ i;++

```

افزودن یک آبجکت ردیف به آرایه‌ی خروجی با کلیدهای خوانا:

اگر تبدیل شد تاریخ استاندارد؛ در غیر این صورت متن خام `.lstart`

```
{
{
{

```

پایان حلقه و شرطهای پارس.

```
echo json_encode(['rows' => $rows], JSON_UNESCAPED_UNICODE);
```

برگرداندن JSON نهایی با کلید `rows` و بدون `escape` یونیکد.

خلاصه عملکرد

ورودی: `limit(٢٠، ١٠٠) تا`، پیش‌فرض `٢٠`.

اجرای یک فرمان `ps` مرتب‌شده بر اساس مصرف CPU (نزوی)، خواندن خروجی متنی.

حذف هدر، پارس هر خط با Regex، استخراج ستون‌ها.

تلash برای نرمال‌سازی زمان شروع پردازه به .Y-m-d H:i:s

اعمال محدودیت تعداد و بازگرداندن JSON

## API

### processes\_list.php:

```
<?php

// api/processes_list.php

header('Content-Type: application/json; charset=utf-8');

$limit = isset($_GET['limit']) ? max(1, min(100, intval($_GET['limit']))) : 20;

/*
 * می‌گیریم: pid, user, comm, %cpu, %mem, lstart
 */
$cmd = "ps -eo pid,user,comm,%cpu,%mem,lstart --sort=-%cpu 2>/dev/null";
$out = @shell_exec($cmd);
$rows = [];

if ($out) {
 $lines = preg_split('/\r?\n/', trim($out));
 if (count($lines) > 0) array_shift($lines); // header

 $i = 0;
```

```

foreach ($lines as $ln) {
 if ($i >= $limit) break;
 // pid user comm cpu mem lstart...
 if (preg_match('/^\s*(\d+)\s+(\S+)\s+(\S+)\s+([\d\.]+)\s+([\d\.]+)\s+(.*)$/', $ln, $m)) {
 $pid = (int)$m[1];
 $user = $m[2];
 $comm = $m[3];
 $cpu = floatval($m[4]);
 $mem = floatval($m[5]);
 $lstr = trim($m[6]);

 $created_at = null;
 $t = @strtotime($lstr);
 if ($t !== false) $created_at = date('Y-m-d H:i:s', $t);

 $rows[] = [
 'pid' => $pid,
 'user' => $user,
 'name' => $comm,
 'cpu_percent' => $cpu,
 'mem_percent' => $mem,
 'created_at' => $created_at ?: $lstr
];
 $i++;
 }
}

```

```
}
```

```
echo json_encode(['rows' => $rows], JSON_UNESCAPED_UNICODE);
```

توضیح:

```
session_start();
```

سشن را شروع می‌کند تا احراز هویت مبتنی بر سشن قابل استفاده باشد.

```
header('Content-Type: application/json; charset=utf-8');
```

نوع پاسخ را JSON با UTF-8 اعلام می‌کند.

```
if (!function_exists('str_starts_with')){
```

```
 function str_starts_with($haystack, $needle){ return $needle!=="" &&
 strpos($haystack,$needle)===0; }
```

```
{
```

اگر تابع str\_starts\_with (نسخه‌های جدید PHP) موجود نباشد، پیاده‌سازی ساده‌ای از آن تعریف می‌کند.

```
function jexit($code, $arr){ http_response_code($code); echo json_encode($arr,
JSON_UNESCAPED_UNICODE); exit; }
```

تابع کمکی: با کد وضعیت HTTP پاسخ JSON را چاپ می‌کند و از اسکریپت خارج می‌شود.

```
function sh($cmd){
```

```
$ des = [1=>'pipe','w'], 2=>'pipe','w'];
$ p = proc_open($cmd, $des, $pipes);
$ out=""; $err=""; $rc=null;
```

```

if(is_resource($p))}

$ out = stream_get_contents($pipes[1]); fclose($pipes[1]);

$ err = stream_get_contents($pipes[2]); fclose($pipes[2]);

$ rc = proc_close($p);

{

return [$rc, $out, $err, $cmd];
}

```

تابع اجرای دستور `shl` با `proc_open`: خروجی استاندارد (`out$`), خطای (`err$`) و کد بازگشت (`rc$`) را می‌گیرد و بر می‌گرداند.

```
function which_first($c){ foreach($c as $x){ if(@is_executable($x)) return $x; } return null; }
```

تابع کمکی: از میان چند مسیر ممکن، اولین فایل اجرایی موجود را بر می‌گرداند.

```

function web_user(){}
if(function_exists('posix_geteuid') && function_exists('posix_getpwuid')){}

$ u = @posix_getpwuid(@posix_geteuid()); if(!empty($u['name'])) return $u['name'];

{
 return getenv('APACHE_RUN_USER') ?: (getenv('USER') ?: 'www-data');
}

```

نام کاربر وب سرور (کاربر فرایند PHP) را تشخیص می‌دهد: ترجیحاً از POSIX، و گرنه از متغیرهای محیطی، پیش‌فرض `www-data`

```
if (empty($_SESSION['uid'])) jexit(401, ['ok'=>false,'error'=>'unauthorized']);
```

اگر کاربر لاگین نیست (سشن uid ندارد) → 401 و خروج.

```
if ($_SERVER['REQUEST_METHOD']!='POST') jexit(405,
['ok'=>false,'error'=>'method_not_allowed']);
```

فقط متد POST مجاز است، غیر از آن → 405 و خروج.

```
$pid = isset($_POST['pid']) ? intval($_POST['pid']) : 0;
if ($pid <= 1) jexit(400, ['ok'=>false, 'error'=>'bad_pid']);
```

پارامتر pid را می‌گیرد، به عدد صحیح تبدیل می‌کند؛ اگر نامعتبر یا خط‌ناک ( $\geq 1$ ) بود → 400

```
$BIN_PS = which_first(['/bin/ps','/usr/bin/ps']) ?: 'ps';
$BIN_KILL = which_first(['/bin/kill','/usr/bin/kill']);
$BIN_SUDO = which_first(['/usr/bin/sudo','/bin/sudo']);
if (!$BIN_KILL) jexit(500, ['ok'=>false,'error'=>'kill_not_found']);
```

مسیر باینری‌های kill، ps و (در صورت وجود) sudo را پیدا می‌کند؛ نبودن kill → 500

```
list($_r1, $uOut, $uErr) = sh("$BIN_PS -o user= -p " . escapeshellarg($pid) . " 2>&1");
list($_r2, $cOut, $cErr) = sh("$BIN_PS -o comm= -p " . escapeshellarg($pid) . " 2>&1");
$user = trim($uOut); $comm = trim($cOut);
if ($user==="" || $comm=="") jexit(404,
['ok'=>false,'error'=>'not_found','stdout'=>$uOut.$cOut,'stderr'=>$uErr.$cErr]);
```

با ps، صاحب پردازه (user) و نام فرمان (comm) را برای PID می‌خواند؛ اگر چیزی برنگشت → 404

```
$deny] =
' systemd','init','kthreadd','rcu_','migration','ksoftirqd','watchdog','cpuhp','kworker','kdevtmpfs',
' sshd','apache2','httpd','nginx','php-fpm','php','mysqld','mariadb',
' dbus-daemon','systemd-udevd','systemd-journald','systemd-logind','systemd-resolved',
' NetworkManager','wpa_supplicant','rsyslogd','cron','crond','containerd','dockerd'
;[
```

```
foreach($deny as $d){ if($comm === $d || str_starts_with($comm,$d)) jexit(403,
['ok'=>false,'error'=>'denied_critical','pid'=>$pid,'user'=>$user,'comm'=>$comm]); }
```

فهرست پردازهای حساس/سیستمی را تعریف می‌کند؛ اگر نام فرمان یکی از آن‌ها (یا با آن شروع شود) باشد → 403 و ممنوع.

```
$WEB = web_user();
```

```
$use_sudo = ($user !== $WEB);
```

تشخیص می‌دهد آیا مالک پردازه با کاربر وب یکی است؛ اگر متفاوت باشد، برای گشتن پردازه نیاز به sudo است.

```
$prefix = $use_sudo && $BIN_SUDO ? "$BIN_SUDO -n;" : "

list($rc1,$out1,$err1,$cmd1) = sh($prefix.$BIN_KILL." -TERM ".escapeshellarg($pid)."
2>&1");

usleep;(250000)

list($_rcA,$aliveOut1,$_errA) = sh("$BIN_PS -p ".escapeshellarg($pid)." -o pid= 2>&1");

$alive1 = trim($aliveOut1);"
```

ابتدا با سیگنال SIGTERM تلاش می‌کند پردازه را مؤدبانه بیندد؛ کمی صبر می‌کند؛ سپس با ps بررسی می‌کند هنوز زنده است یا نه.

```
if ($alive1){

list($rc2,$out2,$err2,$cmd2) = sh($prefix.$BIN_KILL." -KILL ".escapeshellarg($pid)."
2>&1");

usleep;(250000)

list($_rcB,$aliveOut2,$_errB) = sh("$BIN_PS -p ".escapeshellarg($pid)." -o pid= 2>&1");

$alive2 = trim($aliveOut2);"
```

اگر هنوز زنده بود، با SIGKILL (اجباری) می‌کشد؛ دوباره کمی صبر و وضعیت را چک می‌کند.

```
if ($alive2){
```

```

if (strpos($err1.$err2.$out1.$out2, 'password is required') !== false) {
 jexit(403) ,
}

' ok'=>false,'error'=>'needs_sudoers,'

' message'=>'sudo NOPASSWD kill برای sudoers سمت نشده یا محدود است. مسیر درست kill را در sudoers اضافه کن.',

' hint'=>"echo 'www-data ALL=(root) NOPASSWD: /bin/kill, /usr/bin/kill' >
/etc/sudoers.d/iotdash-kill ; visudo -cf /etc/sudoers.d/iotdash-kill,"

' stderr'=>$err1."\n".$err2, 'cmd_term'=>$cmd1, 'cmd_kill'=>$cmd2
;[

{

jexit(500) ,
}

' ok'=>false,'error'=>'kill_failed','pid'=>$pid,'user'=>$user,'comm'=>$comm,
' message'=>'هنوز زنده است,'<='TERM و KILL,'پس از
' stderr'=>trim($err1."\n".$err2), 'stdout'=>trim($out1."\n".$out2)
;[

{
}

```

اگر با KILL هم زنده ماند:

اگر خروجی‌ها نشان دهد sudo پسورد می‌خواهد، پاسخ 403 با پیام روشن درباره تنظیم sudoers می‌دهد؛

.«kill\_failed» با پیام 500 و گرنگ

```
jexit(200, ['ok'=>true,'pid'=>$pid,'user'=>$user,'comm'=>$comm]);
```

اگر پردازه دیگر زنده نیست، پاسخ موفقیت 200 با اطلاعات PID/کاربر/نام فرمان برمی‌گرداند.

#### خلاصه عملکرد

این API پس از احراز هویت و دریافت pid، مسیر ابزارها را می‌یابد، صاحب و نام پردازه را می‌خواند، اگر پردازه «حساس» باشد رد می‌کند، سپس ابتدا SIGTERM و در صورت نیاز SIGKILL می‌فرستد؛ زنده‌بودن را با ps بررسی می‌کند و نتیجه را به صورت JSON با کد وضعیت مناسب برمی‌گرداند.

```
?>php
//api/processes_list.php

header('Content-Type: application/json; charset=utf-8');
```

فایل API برای برگرداندن فهرست پردازه‌ها به صورت JSON

هدر پاسخ را روی application/json با یونیکد تنظیم می‌کند.

```
$limit = isset($_GET['limit']) ? max(1, min(100, intval($_GET['limit']))) : 20;
```

پارامتر limit از کوئری استرینگ خوانده می‌شود.

به عدد صحیح تبدیل، حداقل ۱ و حداکثر ۱۰۰؛ پیش‌فرض ۲۰.

```
*/
```

pid, user, comm, %cpu, %mem, lstart می‌گیریم:

```
/*
```

```
$cmd = "ps -eo pid,user,comm,%cpu,%mem,lstart --sort=-%cpu 2>/dev/null;"
```

دستور سیستم: ps با قالب ستون‌های مشخص:

pid شناسه پردازه

user کاربر مالک

comm نام فرمان (کوتاه)

CPU درصد مصرف %cpu

RAM درصد مصرف %mem

lstart زمان شروع پردازه (فرمت متغیر طولانی)

.CPU-- sort=-%cpu-- یعنی مرتب‌سازی نزولی بر اساس

dev/null/<2 خطاهای استاندارد را دور می‌ریزد تا خروجی تمیز بماند.

```
$out = @shell_exec($cmd);
```

;\$[] = \$rows

دستور اجرا می‌شود و خروجی کامل به صورت رشته در \$out می‌آید. @ خطاهای احتمالی را خاموش می‌کند.

آرایه‌ی نهایی ردیف‌ها را در \$rows جمع می‌کند.

```

if ($out)

$ lines = preg_split('/\r?\n/', trim($out));

if (count($lines) > 0) array_shift($lines); // header

```

اگر خروجی خالی نبود:

خروجی به خطوط جدا می‌شود.

اولین خط هدر ps است؛ حذف می‌شود.

```

$ i = 0;

foreach ($lines as $ln)

if ($i >= $limit) break;

```

شمارنده‌ی تعداد ردیف‌های اضافه شده.

اگر به سقف limit رسید، حلقه شکسته می‌شود.

// pid user comm cpu mem lstart...

```
if (preg_match('/^(\d+)\s+(\S+)\s+(\S+)\s+(\d\.\d+)\s+(.*$)', $ln, $m)) {
```

هر خط با یک عبارت منظم (Regex) پارس می‌شود:

$(\d+)$  → PID

$(\S+)$  → USER

(\S+) → COMM

([\d\.]\*) → %CPU

([\d\.]\*) → %MEM

→ بقیه‌ی خط = lstart به صورت متن (ممکن است فاصله داشته باشد) (\*.)

```
$ pid = (int)$m[1]
$ user = $m[2]
$ comm = $m[3]
$ cpu = floatval($m[4]);
$ mem = floatval($m[5]);
$ lstr = trim($m[6]);
```

نگاشت گروه‌های Regex به متغیرها و تبدیل‌های عددی لازم.

```
$ created_at = null;
$t = @strtotime($lstr);
if ($t !== false) $created_at = date('Y-m-d H:i:s', $t);
```

تلاش برای تبدیل lstart متنی به timestamp با .strtotime

اگر موفق بود، به فرمت استاندارد Y-m-d H:i:s تبدیل می‌کند؛ و گرنه همان رشته‌ی اصلی بعداً استفاده می‌شود.

```
$ rows] = []
' pid' => $pid,
```

```

' user' => $user,
' name' => $comm,
' cpu_percent' => $cpu,
' mem_percent' => $mem,
' created_at' => $created_at ?: $lstr
;[
$ i;++

```

افزودن یک آبجکت ردیف به آرایه‌ی خروجی با کلیدهای خوانا:

اگر تبدیل شد تاریخ استاندارد؛ در غیر این صورت متن خام `.lstart`

```
{
{
{

```

پایان حلقه و شرطهای پارس.

```
echo json_encode(['rows' => $rows], JSON_UNESCAPED_UNICODE);
```

برگرداندن JSON نهایی با کلید `rows` و بدون `escape` یونیکد.

خلاصه عملکرد

ورودی: `limit(٢٠، ١٠٠) تا`، پیش‌فرض `٢٠`.

اجرای یک فرمان `ps` مرتب‌شده بر اساس مصرف CPU (نزوی)، خواندن خروجی متنی.

حذف هدر، پارس هر خط با Regex، استخراج ستون‌ها.

تلash برای نرمال‌سازی زمان شروع پردازه به .Y-m-d H:i:s

اعمال محدودیت تعداد و بازگرداندن JSON

## API

### system\_health.php:

```
<?php

// api/system_health.php

header('Content-Type: application/json; charset=utf-8');

/** CPU% /proc/stat کوتاه از با نمونه برداری */

function read_cpu_stat() {
 $s = @file('/proc/stat');
 if(!$s) return null;
 foreach($s as $line){
 if(strpos($line, 'cpu ') === 0){
 $parts = preg_split('/\s+/', trim($line));
 return array_map('intval', array_slice($parts, 1, 10));
 }
 }
 return null;
}
```

```

function cpu_percent(){

$a = read_cpu_stat();

if(!$a){ return null; }

usleep(200000); // 0.2s

$b = read_cpu_stat();

if(!$b){ return null; }

$idleA = $a[3] + $a[4];

$idleB = $b[3] + $b[4];

$totalA = array_sum($a);

$totalB = array_sum($b);

$totald = $totalB - $totalA;

$idled = $idleB - $idleA;

if ($totald <= 0) return null;

$usage = (1.0 - ($idled / $totald)) * 100.0;

return max(0.0, min(100.0, $usage));

}

```

```

/** RAM% ;!/proc/meminfo **/

function ram_percent(){

$mi = @file('/proc/meminfo');

if(!$mi) return null;

$data = [];

foreach($mi as $ln){

if (preg_match('/^(\w+):\s+(\d+)/', $ln, $m)){

$data[$m[1]] = (int)$m[2]; // kB

}

```

```

}

if(empty($data['MemTotal'])) return null;

$memTotal = $data['MemTotal'];

$memAvail = isset($data['MemAvailable']) ? $data['MemAvailable'] : ($data['MemFree'] ?? 0);

$used = max(0, $memTotal - $memAvail);

return ($used / $memTotal) * 100.0;

}

```

/\*\* دمای CPU \*\*/

```

function cpu_temp_c(){

$cands = glob('/sys/class/thermal/thermal_zone*/temp');

if(!$cands) return null;

foreach($cands as $p){

$raw = @file_get_contents($p);

if($raw !== false){

$v = (int)trim($raw);

if($v > 1000) return $v / 1000.0;

return floatval($v);

}

}

return null;

}

```

/\*\* آدرس IP اولین (غیرلوكال IPv4) \*\*/

```

function ip_addr(){

```

```

$out = @shell_exec('LC_ALL=C hostname -I 2>/dev/null');

if($out){

 $parts = preg_split('/\s+/', trim($out));

 foreach($parts as $ip){

 if (filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4) && $ip !== '127.0.0.1'){

 return $ip;

 }

 }

}

return $_SERVER['SERVER_ADDR'] ?? null;

}

```

```

/** كشف اينترفيسيس وايرلس مثلًا (wlan0) **/

function detect_wlan_iface(){

$lines = [];

@exec('LC_ALL=C iw dev 2>/dev/null', $lines);

foreach($lines as $ln){

if (preg_match('/^s*Interface\s+(\S+)/i', $ln, $m)){

 return trim($m[1]);

}

}

// fallback

return 'wlan0';

}

```

/\*\* اطلاعات Wi-Fi از iw (بیشنهادی) \*\*/

```

function wifi_info(){
 $iface = detect_wlan_iface();
 // اگر چنین اینترفیسی نداریم، بدء null
 if(!$iface) return null;

 $ssid = null; $rss = null; $bitrate = null; $freq = null; $connected = false;

 $out = [];
 @exec("LC_ALL=C iw dev {$iface} link 2>/dev/null", $out);
 if ($out) {
 foreach($out as $ln){
 if (stripos($ln, 'Not connected') !== false) { $connected = false; break; }
 if (preg_match('/SSID:\s*(.+)\$/i', $ln, $m)) { $ssid = trim($m[1]); $connected = true; }
 if (preg_match('/signal:\s*(-?\d+)\s*dBm/i', $ln, $m)) { $rss = intval($m[1]); }
 if (preg_match('/tx bitrate:\s*([0-9\.]+)\s*\S+/\i', $ln, $m)) { $bitrate = trim($m[1]); }
 if (preg_match('/freq:\s*([0-9]+)\/i', $ln, $m)) { $freq = intval($m[1]); }
 }
 }

 // dBm: -90→0% ، -30→100% کیفیت تقریبی از
 $quality = null;
 if ($rss != null) {
 $quality = max(0, min(100, (int)(($rss + 90) * (100.0/60.0))));
 }
}

```

```

// همان اینترفیس وای فای IP
$ip = null;
$ipa = [];

@exec("ip -4 addr show {$iface} 2>/dev/null | awk '/inet / {print \$2}' | cut -d/ -f1 | head -n1",
$ipa);

if(isset($ipa[0])) $ip = trim($ipa[0]);

```

را از آن هم امتحان می کنیم RSSI هست، جهت سازگاری iwconfig در دسترس نبود ولی Wi-Fi اگر //

```

if ($rss == null) {

 $iwc = @shell_exec('LC_ALL=C iwconfig 2>/dev/null');

 if($iwc && preg_match('/Signal level[=:]\\s*(-?\\d+)\\s*dBm/i', $iwc, $m)) {

 $rss = (int)$m[1];

 $quality = max(0, min(100, (int)(($rss + 90) * (100.0/60.0))));

 }

}

```

```

return [
 'iface' => $iface,
 'connected'=> $connected,
 'ip' => $ip,
 'ssid' => $ssid,
 'rssi_dbm' => $rss, // 54- مثلًا
 'quality' => $quality, // 75 مثلًا
 'bitrate' => $bitrate, // 72.2" MBit/s مثلًا
 'freq' => $freq // 2437 مثلًا
]

```

```

];
}

/** فرستاد wifi_rssi برای سازگاری با کد فعلی تو که می
function wifi_rssi_compat(){

$w = wifi_info();

return $w ? $w['rss_i_dbm'] : null;
}

$cpu = cpu_percent();

$ram = ram_percent();

$temp= cpu_temp_c();

$ip = ip_addr();

$wifi= wifi_info();

$rssi= wifi_rssi_compat();

echo json_encode([
'online' => true,
'cpu_percent' => $cpu !== null ? round($cpu,1) : null,
'ram_percent' => $ram !== null ? round($ram,1) : null,
'cpu_temp_c' => $temp !== null ? round($temp,1) : null,
'ip_addr' => $ip,
'wifi_rssi' => $rssi, // برای کدهای قدیمی فراتاند
'wifi' => $wifi // آجکت کامل برای نمایش پیشرفت
], JSON_UNESCAPED_UNICODE|JSON_UNESCAPED_SLASHES);

```

توضیح :

```
<?php
// api/system_health.php
header('Content-Type: application/json; charset=utf-8');
```

فایل API برای گزارش سلامت سیستم؛ خروجی JSON با یونیکد.

```
خواندن مصرف CPU از /proc/stat

function read_cpu_stat() {
 $s = @file('/proc/stat');
 if(!$s) return null;
 foreach($s as $line){
 if(strpos($line, 'cpu ') === 0){
 $parts = preg_split('/\s+/', trim($line));
 return array_map('intval', array_slice($parts, 1, 10));
 }
 }
 return null;
}
```

را خط به خط می خواند.

اولین خطی که با 'cpu' شروع می شود را می گیرد (تجمیعی کل هسته ها).

اعداد بعد از int cpu (user, nice, system, idle, iowait, irq, softirq, ...) تبدیل و برمی گرداند.

اگر فایل/خط در دسترس نبود : null

```
function cpu_percent(){
 $a = read_cpu_stat();
 if(!$a){ return null; }
 usleep(200000); // 0.2s
 $b = read_cpu_stat();
 if(!$b){ return null; }
 $idleA = $a[3] + $a[4];
 $idleB = $b[3] + $b[4];
 $totalA = array_sum($a);
 $totalB = array_sum($b);
 $totald = $totalB - $totalA;
 $idled = $idleB - $idleA;
 if ($totald <= 0) return null;
 $usage = (1.0 - ($idled / $totald)) * 100.0;
 return max(0.0, min(100.0, $usage));
}
```

دو نمونه برداری از شمارنده‌های CPU با فاصله‌ی 0.2 ثانیه.

اختلاف کل زمان (totald) و اختلاف زمان بیکار (idled) را می‌گیرد.

- idled/totald × 100.1 = درصد استفاده

مقدار را در بازه [0..100] نگه می‌دارد. خطای غیرعادی. → null

خواندن مصرف RAM از /proc/meminfo

```
function ram_percent(){
$mi = @file('/proc/meminfo');
if(!$mi) return null;
$data = [];
foreach($mi as $ln){
if (preg_match('/^(\w+):\s+(\d+)/', $ln, $m)){
$data[$m[1]] = (int)$m[2]; // kB
}
}
if(empty($data['MemTotal'])) return null;
$memTotal = $data['MemTotal'];
$memAvail = isset($data['MemAvailable']) ? $data['MemAvailable'] : ($data['MemFree'] ?? 0);
$used = max(0, $memTotal - $memAvail);
return ($used / $memTotal) * 100.0;
}
```

را بر حسب کیلوبایت می خواند (fallback: MemFree .MemAvailable MemTotal).

$$\text{درصد استفاده} = (\text{Total} - \text{Available}) / \text{Total} \times 100.$$

دماي CPU sysfs از

```
function cpu_temp_c(){
$cands = glob('/sys/class/thermal/thermal_zone*/temp');
if(!$cands) return null;
```

```

foreach($cands as $p){

 $raw = @file_get_contents($p);

 if($raw !== false){

 $v = (int)trim($raw);

 if($v > 1000) return $v / 1000.0;

 return floatval($v);

 }

}

return null;
}

```

مسیرهای دمای حرارتی را لیست می‌کند.

اولین مقدار معتبر را می‌خواند.

اگر عدد بزرگ‌تر از 1000 بود بر حسب میلی‌درجه است → تقسیم بر 1000 (درجه‌ی سانتی‌گراد).

آدرس IP دستگاه

```

function ip_addr(){

 $out = @shell_exec('LC_ALL=C hostname -I 2>/dev/null');

 if($out){

 $parts = preg_split('/\s+/', trim($out));

 foreach($parts as $ip){

 if(filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4) && $ip !== '127.0.0.1'){

 return $ip;

 }

 }

 }

}

```

```

 }
}

return $_SERVER['SERVER_ADDR'] ?? null;
}

```

خروجی - hostname را می‌گیرد؛ بین IP ها اولین IPv4 غیرلوکال ( $\neq 127.0.0.1$ ) را برمی‌گرداند.

اگر نشد: از `$_SERVER['SERVER_ADDR']` استفاده می‌کند (ممکن است خالی باشد).

کشف نام اینترفیس وای‌فای

```

function detect_wlan_iface(){

$lines = [];

@exec('LC_ALL=C iw dev 2>/dev/null', $lines);

foreach($lines as $ln){

if (preg_match('/^Interface\s+(\S+)/i', $ln, $m)){
 return trim($m[1]);
}

}

// fallback

return 'wlan0';
}

```

خروجی `iw dev` را می‌خواند؛ اولین `Interface <name>` را برمی‌گرداند.

اگر پیدا نشد: پیش‌فرض `'wlan0'`.

اطلاعات) Wi-Fi با + سازگاری با `iwconfig`

```

function wifi_info(){
 $iface = detect_wlan_iface();
 if(!$iface) return null;

 $ssid = null; $rssi = null; $bitrate = null; $freq = null; $connected = false;

 $out = [];
 @exec("LC_ALL=C iw dev {$iface} link 2>/dev/null", $out);
 if ($out) {
 foreach($out as $ln){
 if (strpos($ln, 'Not connected') !== false) { $connected = false; break; }

 if (preg_match('/SSID:\s*(.+)\$/i', $ln, $m)) { $ssid = trim($m[1]); $connected = true; }

 if (preg_match('/signal:\s*(-?\d+)\s*dBm/i', $ln, $m)) { $rssi = intval($m[1]); }

 if (preg_match('/tx bitrate:\s*([0-9\.]+)\s*/i', $ln, $m)) { $bitrate = trim($m[1]); }

 if (preg_match('/freq:\s*([0-9]+)\s*/i', $ln, $m)) { $freq = intval($m[1]); }
 }
 }
}

;iw dev <iface> link SSID .RSSI (dBm) ، نرخ ارسال (bitrate) و فرکانس را استخراج می کند

```

اگر «Not connected» دیده شود. «connected=false»

// dBm: -90→0% ، -30→100% کیفیت تقریبی از

```

$quality = null;
if ($rssi != null) {
 $quality = max(0, min(100, (int)(($rssi + 90) * (100.0/60.0))));
}

```

```
}
```

تخمین کیفیت سیگنال از RSSI نگاشت خطی از بازه [-90..-30] dBm به [100..0] درصد.

```
// همان اینترفیس وای فای IP
```

```
$ip = null;
```

```
$ipa = [];
```

```
@exec("ip -4 addr show {$iface} 2>/dev/null | awk '/inet /{print \$2}' | cut -d/ -f1 | head -n1",
$ipa);
```

```
if(isset($ipa[0])) $ip = trim($ipa[0]);
```

اولین IPv4 اختصاص یافته به همان اینترفیس وای فای را استخراج می کند.

```
// را از آن هم امتحان می کنیم RSSI در دسترس نبود ولی iwconfig هست، جهت سازگاری RSSI در دسترس نبود ولی اگر
```

```
if ($rss == null) {
```

```
$iwc = @shell_exec('LC_ALL=C iwconfig 2>/dev/null');
```

```
if($iwc && preg_match('/Signal level[:]\s*(-?\d+)\s*dBm/i', $iwc, $m)) {
```

```
$rss = (int)$m[1];
```

```
$quality = max(0, min(100, (int)(($rss + 90) * (100.0/60.0))));
```

```
}
```

```
}
```

اگر iwconfig نیامد یا RSSI نبود تلاش دوم برای گرفتن RSSI هست:

```
return [
```

```
'iface' => $iface,
```

```
'connected'=> $connected,
```

```
'ip' => $ip,
```

```

'ssid' => $ssid,
'rss_i_dbm' => $rss_i,
'quality' => $quality,
'bitrate' => $bitrate,
'freq' => $freq
];
}

```

آجکت کامل Wi-Fi برمی‌گرداند.

```

لایه‌ی سازگاری با کدهای قدیمی

function wifi_rssi_compat(){
$w = wifi_info();
return $w ? $w['rss_i_dbm'] : null;
}

```

برای خروجی قدیمی که فقط wifi\_rssi می‌خواست، صرفاً dBm را برمی‌گرداند.

```

گردآوری، گردکردن، و خروجی JSON

$cput = cpu_percent();
$ram = ram_percent();
$temp = cpu_temp_c();
$ip = ip_addr();
$wifi = wifi_info();
$rssi = wifi_rssi_compat();

echo json_encode([

```

```
'online' => true,
'cpu_percent' => $cpu !== null ? round($cpu,1) : null,
'ram_percent' => $ram !== null ? round($ram,1) : null,
'cpu_temp_c' => $temp !== null ? round($temp,1) : null,
'ip_addr' => $ip,
'wifi_rssi' => $rss, // سازگار با فرانتاند قدیمی
'wifi' => $wifi // آبجکت کامل برای نمایش پیشرفته
```

], JSON\_UNESCAPED\_UNICODE|JSON\_UNESCAPED\_SLASHES);

مقادیر را می‌گیرد؛ اعداد را تا ۱ رقم اعشار گرد می‌کند.

online=true به عنوان پرچم سلامت.

wifi\_rssi قدیمی و هم آبجکت wifi جدید را برمی‌گرداند.

يونیکد و اسلش‌ها را escape نمی‌کند تا خروجی خواناتر باشد.

خلاصه‌ی عملکرد

این API، وضعیت لحظه‌ای دستگاه را از سیستم‌فایل‌ها و ابزارهای سیستمی می‌خواند

/proc/stat از CPU% با نمونه‌برداری کوتاه

/proc/meminfo از RAM%

از CPU دمای /sys/class/thermal/.../temp

IP از - ایمومی

وضعیت وایفای (SSID، RSSI (dBm)، کیفیت %، فرکانس، bitrate) و آنترفیس (از IP)

خروجی نهایی JSON است؛ شامل cpu\_percent، ram\_percent، cpu\_temp\_c، ip\_addr، wifi\_rssi، wifi کامل

API

system\_power.php:

```
<?php

// api/system_power.php

header('Content-Type: application/json; charset=utf-8');

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
 http_response_code(405);
 echo json_encode(['ok'=>false, 'error'=>'Method not allowed']);
 exit;
}

$action = $_POST['action'] ?? '';
$allowed = ['reboot','shutdown'];
if (!in_array($action, $allowed, true)) {
 http_response_code(400);
 echo json_encode(['ok'=>false, 'error'=>'Invalid action']);
 exit;
}
```

```
}
```

```
// برای امنیت: فقط این دو باینری مجاز
$cmd = ($action === 'reboot')
 ? '/sbin/reboot'
 : '/sbin/poweroff'; // یا: /sbin/shutdown -h now
```

```
// اجرای بدون بلاک کردن درخواست HTTP
```

```
$cmd_full = sprintf('sudo %s > /dev/null 2>&1 &', $cmd);
$ret = null; $out = null;
@exec($cmd_full, $out, $ret);

if ($ret === 0) {
 echo json_encode(['ok'=>true, 'action'=>$action]);
} else {
 http_response_code(500);
 echo json_encode(['ok'=>false, 'error'=>'Command failed', 'code'=>$ret]);
}
```

توضیح :

## ۱ هدر خروجی

```
header('Content-Type: application/json; charset=utf-8');
```

پاسخ را به صورت **JSON** با **UTF-8** برمی‌گرداند.

## ۲ بررسی متدهای درخواست

```

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {

 http_response_code(405);

 echo json_encode(['ok'=>false, 'error'=>'Method not allowed']);

 exit;

}

```

• فقط درخواست‌های **POST** پذیرفته می‌شوند.

• اگر متده دیگری بود → خطای (Method Not Allowed) 405 و خروج.

### 3 گرفتن پارامتر **action**

```

$action = $_POST['action'] ?? "";

$allowed = ['reboot','shutdown'];

if (!in_array($action, $allowed, true)) {

 http_response_code(400);

 echo json_encode(['ok'=>false, 'error'=>'Invalid action']);

 exit;

}

```

• متغیر **action** از ورودی POST خوانده می‌شود.

• فقط دو مقدار معتبرند **reboot** یا **shutdown**.

• اگر غیر از این بود → خطای (Bad Request).400

### 4 انتخاب دستور سیستمی

```

$cmd = ($action === 'reboot')

? '/sbin/reboot'

: '/sbin/poweroff'; // یا: /sbin/shutdown -h now

```

- اگر کاربر reboot بخواهد → دستور /sbin/reboot.
- اگر دستور shutdown -h now می‌توانست /sbin/poweroff یا می‌توانست /sbin/shutdown بخواهد → دستور shutdown.
- فقط همین باینری‌ها مجاز هستند تا امنیت بیشتر باشد.

## ۱۵ جرای دستور

```
$cmd_full = sprintf('sudo %s > /dev/null 2>&1 &', $cmd);
$ret = null; $out = null;
@exec($cmd_full, $out, $ret);
```

- دستور کامل ساخته می‌شود sudo [command] (&) و در پس زمینه اجرا می‌شود.
- خروجی استاندارد و خطای dev/null هدایت می‌شوند تا پاسخی در HTTP ندهد.
- اجرا می‌کند و کد بازگشت را در \$ret ذخیره می‌کند.

## ۶ بررسی نتیجه

```
if ($ret === 0) {
 echo json_encode(['ok'=>true, 'action'=>$action]);
} else {
 http_response_code(500);
 echo json_encode(['ok'=>false, 'error'=>'Command failed', 'code'=>$ret]);
}
```

- اگر دستور با موفقیت اجرا شد ( $$ret == 0$ ) → JSON پاسخ مثبت با ok=true.
- اگر خطای داد → کد وضعیت 500 + پیام خطای دستور.

## خلاصه عملکرد

این API یک کلید کنترل سیستم است:

- فقط متد POST و اکشن‌های shutdown یا reboot مجاز ند.
- برای امنیت فقط باینری‌های مشخص (/sbin/reboot) و (/sbin/poweroff) و (/sbin/poweroff) اجرا می‌شوند.
- با sudo در پس‌زمینه اجرا می‌شود تا ارتباط HTTP قطع نشود.
- در نهایت نتیجه به شکل JSON (موفق یا خطأ) برگردانده می‌شود.

## API

system\_spec.php:

```
<?php

// api/system_specs.php

header('Content-Type: application/json; charset=utf-8');

function read_first_line($path){

 $s = @file_get_contents($path);

 return $s ? trim($s) : null;

}

function cpu_model(){

 $txt = @file_get_contents('/proc/cpuinfo');

 if($txt){

 foreach(['model name','Hardware','Processor'] as $k){

 if (preg_match('/^:'.$k.'\s*:\s*(.+)$mi', $txt, $m)){

 return trim($m[1]);

 }

 }

 }

}
```

```

 return null;
}

function cpu_cores(){
 $n = @shell_exec('nproc 2>/dev/null');

 if($n) return (int)trim($n);
 return null;
}

function board_model(){
 $m = read_first_line('/proc/device-tree/model');

 if($m) return $m;

 $osr = @file_get_contents('/etc/os-release');

 if($osr && preg_match('/PRETTY_NAME="([^"]+)"/', $osr, $mm)){
 return $mm[1];
 }

 return 'Orange Pi (unknown model)';
}

function os_pretty(){
 $osr = @file_get_contents('/etc/os-release');

 if($osr && preg_match('/PRETTY_NAME="([^"]+)"/', $osr, $m)) return $m[1];

 return php_uname('s').'.'.php_uname('r');
}

function kernel(){ return php_uname('r'); }

```

```

function arch(){ return php_uname('m'); }

function hostname_(){ $h = @php_uname('n'); return $h ?: @gethostname(); }

function mem_info(){

// از MemAvailable دقیق‌تر از استفاده می‌کنیم (MemFree + Cached + Buffers)

$total_kb = $avail_kb = null;

$mi = @file('/proc/meminfo');

if($mi){

foreach($mi as $ln){

if (preg_match('/^MemTotal:\s+(\d+)\s*kB/i', $ln, $m)) $total_kb = (int)$m[1];

if (preg_match('/^MemAvailable:\s+(\d+)\s*kB/i', $ln, $m)) $avail_kb = (int)$m[1];

}

}

if($total_kb){

$total_mb = round($total_kb/1024.0, 1);

if($avail_kb !== null){

$free_mb = round($avail_kb/1024.0, 1);

$used_mb = max(0, round($total_mb - $free_mb, 1));

return ['total_mb'=>$total_mb, 'used_mb'=>$used_mb, 'free_mb'=>$free_mb];

}

// fallback: اگر MemAvailable نبود

$free_mb = $used_mb = null;

return ['total_mb'=>$total_mb, 'used_mb'=>$used_mb, 'free_mb'=>$free_mb];

}

return ['total_mb'=>null, 'used_mb'=>null, 'free_mb'=>null];

}

```

```

function disk_info(){

 $total = @disk_total_space('/');
 $free = @disk_free_space('/');

 if($total && $free != false){

 $total_mb = round($total/1024/1024, 1);
 $free_mb = round($free/1024/1024, 1);

 $used_mb = max(0, round($total_mb - $free_mb, 1));

 return ['total_mb'=>$total_mb,'used_mb'=>$used_mb,'free_mb'=>$free_mb];
 }

 return ['total_mb'=>null,'used_mb'=>null,'free_mb'=>null];
}

$mem = mem_info();

$disk = disk_info();

echo json_encode([
 'device_model' => board_model(),
 'hostname' => hostname_(),
 'os' => os_pretty(),
 'kernel' => kernel(),
 'arch' => arch(),
 'cpu_model' => cpu_model(),
 'cpu_cores' => cpu_cores(),
 'mem_total_mb' => $mem['total_mb'],
 'mem_used_mb' => $mem['used_mb'],
]);

```

```
'mem_free_mb' => $mem['free_mb'],
'disk_total_mb' => $disk['total_mb'],
'disk_used_mb' => $disk['used_mb'],
'disk_free_mb' => $disk['free_mb'],
], JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT);
```

توضیح :

```
?>php
//api/system_specs.php

header('Content-Type: application/json; charset=utf-8');
```

فایل API برای برگرداندن «مشخصات سیستم» به صورت JSON

هدر پاسخ: application/json با یونیکد.

ابزار کمکی: خواندن خط اول یک فایل

```
function read_first_line($path){
 $ s = @file_get_contents($path);
 return $s ? trim($s) : null;
}
```

سعی می‌کند کل فایل را بخواند و نتیجه را trim می‌کند.

اگر نشد (خطا یا نبودن فایل): null

نکته: برای فایل‌های خیلی بزرگ، بهتر است به جای fgets یک خطی استفاده شود؛ اینجا فایل‌های هدف کوچک‌اند.

مدل CPU از proc/cpuinfo/

```
function cpu_model() {
 $ txt = @file_get_contents('/proc/cpuinfo');
 if($txt)
 foreach(['model name','Hardware','Processor'] as $k)
 if (preg_match('/^'.$k.':\s*(.+)\$/mi', $txt, $m))
 return trim($m[1]);
 }
 {
 {
 return null;
 }
}
```

محتوای proc/cpuinfo/ را می‌خواند.

به ترتیب دنبال یکی از کلیدهای رایج می‌گردد: .model name (x86), Hardware/Processor (ARM)

اولین مقدار مطابق را برمی‌گرداند؛ وگرنه null.

فلگ‌های m و i در Regex به معنی چندخطی و غیرحساس به حروف.

تعداد هسته‌ها

```

function cpu_cores{()
$ n = @shell_exec('nproc 2>/dev/null');
if($n) return (int)trim($n);
return null;
}

```

با دستور `nproc` تعداد هسته‌های پردازشی را می‌خواند.

خروجی به عدد صحیح تبدیل می‌شود؛ اگر در دسترس نبود: `null`.

### مدل برد/دستگاه

```

function board_model{()
$ m = read_first_line('/proc/device-tree/model');
if($m) return $m;
$ osr = @file_get_contents('/etc/os-release');
if($osr && preg_match('/PRETTY_NAME="([^\"]+)/', $osr, $mm)) {
 return $mm[1]
}
return 'Orange Pi (unknown model);'
}

```

ابتدا تلاش می‌کند مدل برد را از `/proc/device-tree/model` متداول ARM/Embedded بگیرد (روی دستگاه‌های استفاده می‌کند).  
است).

اگر نبود، از `PRETTY_NAME` در `/etc/os-release` استفاده می‌کند.

اگر هیچ کدام نبود: مقدار پیش فرض توصیفی.

نام زیبا/انسان خوان OS

```
function os_pretty(){
 $ osr = @file_get_contents('/etc/os-release');
 if($osr && preg_match('/PRETTY_NAME="([^"]+)"/', $osr, $m)) return $m->[1]
 return php_uname('s').'.'.php_uname('r');
}
```

به نام سیستم و نسخه‌ی کرنل از fallback را برمی‌گرداند؛ در غیر این صورت PRETTY\_NAME باشد،  
php\_uname.

نسخه‌ی کرنل، معماری، میزبان

```
function kernel(){ return php_uname('r'); }
function arch(){ return php_uname('m'); }
function hostname_(){ $h = @php_uname('n'); return $h ?: @gethostname ;();}
kernel: نسخه‌ی کرنل (مثل 5.15.0...)
```

arch: معماری (مثل x86\_64, armv7l, aarch64)

hostname: نام میزبان؛ اول از ('n') php\_uname و در صورت خالی بودن از .()gethostname

(MB) حافظه اطلاعات

```
function mem_info(){
 MemAvailable // از استفاده می‌کنیم (دقیق‌تر از MemFree + Cached + Buffers(
```

```

$ total_kb = $avail_kb = null;
$ mi = @file('/proc/meminfo');
if($mi)
foreach($mi as $ln)
if (preg_match('/^MemTotal:\s+(\d+)\s*kB/i', $ln, $m)) $total_kb = (int)$m;[1]
if (preg_match('/^MemAvailable:\s+(\d+)\s*kB/i', $ln, $m)) $avail_kb = (int)$m;[1]
{
{
if($total_kb)
$ total_mb = round($total_kb/1024.0, 1);
if($avail_kb !== null)
$ free_mb = round($avail_kb/1024.0, 1);
$ used_mb = max(0, round($total_mb - $free_mb, 1));
return ['total_mb'=>$total_mb, 'used_mb'=>$used_mb, 'free_mb'=>$free_mb];
{
// fallback اگر MemAvailable نبود
$ free_mb = $used_mb = null;
return ['total_mb'=>$total_mb, 'used_mb'=>$used_mb, 'free_mb'=>$free_mb];
{
return ['total_mb'=>null, 'used_mb'=>null, 'free_mb'=>null];
{
 . . .

```

proc/meminfo/ را می‌خواند و MemAvailable و ترجیحاً MemTotal را استخراج می‌کند.

به MB تبدیل کرده و برمی‌گرداند: کل، مصرف شده، آزاد.

اگر `MemAvailable` موجود نباشد: فقط کل را می‌دهد و بقیه `null`.

اطلاعات دیسک ریشه (MB)

```
function disk_info(){
 $ total = @disk_total_space;()
 $ free = @disk_free_space;()

 if($total && $free !== false){

 $ total_mb = round($total/1024/1024, 1);
 $ free_mb = round($free/1024/1024, 1);
 $ used_mb = max(0, round($total_mb - $free_mb, 1));

 return ['total_mb'=>$total_mb,'used_mb'=>$used_mb,'free_mb'=>$free_mb];
 }

 return ['total_mb'=>null,'used_mb'=>null,'free_mb'=>null];
}
```

فضای کل/آزاد پارتیشن / را از توابع داخلی PHP می‌گیرد.

به MB تبدیل و مصرف شده را محاسبه می‌کند.

اگر خواندن ممکن نبود: مقادیر `null`.

گردآوری نهایی و خروجی JSON

```
$mem = mem_info();
$disk = disk_info();
```

```

echo json_encode()]

' device_model' => board_model,()

' hostname' => hostname,()

' os' => os.pretty,()

' kernel' => kernel,()

' arch' => arch,()

' cpu_model' => cpu_model,()

' cpu_cores' => cpu_cores,()

' mem_total_mb' => $mem['total_mb'],

' mem_used_mb' => $mem['used_mb'],

' mem_free_mb' => $mem['free_mb'],

' disk_total_mb' => $disk['total_mb'],

' disk_used_mb' => $disk['used_mb'],

' disk_free_mb' => $disk['free_mb'],

,[JSON_UNESCAPED_UNICODE]JSON_PRETTY_PRINT;()

```

توابع را صدا می‌زند و یک آبجکت JSON تمیز بر می‌گرداند:

مشخصات دستگاه/OS/کرنل/معماری/.CPU

تعداد هسته‌ها،

وضعیت حافظه و دیسک (MB).

گزینه‌ها:

.uXXXX\ → متن فارسی بدون JSON\_UNESCAPED\_UNICODE

→ خوانایی بیشتر. JSON\_PRETTY\_PRINT

خلاصه‌ی عملکرد

API مشخصات سیستم را مستقیم از فایل‌های مجازی لینوکس (/proc, /sys, /etc/os-release) و توابع سیستم می‌خواند؛ دیتابیس/اکویری/ایندکس ندارد.

خروجی شامل مدل برد/CPU، تعداد هسته‌ها، نام میزبان، نسخه‌ی OS/کرنل/معماری، و آمار حافظه/دیسک به MB است.

وابستگی‌ها: وجود /proc/\* و /sys/\* (روی لینوکس آرم موجود)، و دسترسی خواندن این مسیرها توسط PHP

## API

### \_kill\_diag.php:

```
<?php

// api/_kill_diag.php

session_start();

header('Content-Type: application/json; charset=utf-8');

function j($x){ echo json_encode($x, JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT);
exit; }

function sh($cmd){

$des = [1=>'pipe','w'], 2=>'pipe','w']];
```

```

$p = proc_open($cmd, $des, $pipes);

$out=""; $err=""; $rc=null;

if(is_resource($p)){

 $out = stream_get_contents($pipes[1]); fclose($pipes[1]);

 $err = stream_get_contents($pipes[2]); fclose($pipes[2]);

 $rc = proc_close($p);

}

return ['rc'=>$rc,'out'=>$out,'err'=>$err,'cmd'=>$cmd];

}

function which($cands){ foreach($cands as $p){ if(@is_executable($p)) return $p; } return null; }

$pid = isset($_REQUEST['pid']) ? (int)$_REQUEST['pid'] : 0;

$webuser = function_exists('posix_geteuid') && function_exists('posix_getpwuid')
? (posix_getpwuid(posix_geteuid())['name'] ?? 'www-data')
: (getenv('APACHE_RUN_USER') ?: (getenv('USER') ?: 'www-data'));

$BIN_PS = which(['/bin/ps','/usr/bin/ps']) ?: 'ps';
$BIN_KILL = which(['/bin/kill','/usr/bin/kill']);
$BIN_SUDO = which(['/usr/bin/sudo','/bin/sudo']);

$info = [
 'session_uid' => $_SESSION['uid'] ?? null,
 'web_user' => $webuser,
 'paths' => [

```

```

'ps' => $BIN_PS,
'kill' => $BIN_KILL,
'sudo' => $BIN_SUDO,
],
'sudo_nopass_check' => $BIN_SUDO ? sh("$BIN_SUDO -n true 2>&1") : null,
];

```

```

if($pid > 0){
$pUser = sh("$BIN_PS -o user= -p $pid 2>&1");
$pComm = sh("$BIN_PS -o comm= -p $pid 2>&1");
$info['process'] = ['pid'=>$pid, 'ps_user'=>$pUser, 'ps_comm'=>$pComm];

```

// فقط اگر پارامتر KILL و TERM تست واقعی بدهی

```

if(isset($_REQUEST['do']) && $_REQUEST['do']=='1'){
if(!$BIN_KILL){ j(['ok'=>false, 'error'=>'kill_not_found', 'diag'=>$info]); }
$term = sh(($BIN_SUDO ? "$BIN_SUDO -n \"\"")."$BIN_KILL -TERM $pid 2>&1");
usleep(250000);
$alive1 = trim(sh("$BIN_PS -p $pid -o pid= 2>&1")['out']) != "";
$kill = null; $alive2 = null;
if($alive1){
$kill = sh(($BIN_SUDO ? "$BIN_SUDO -n \"\"")."$BIN_KILL -KILL $pid 2>&1");
usleep(250000);
$alive2 = trim(sh("$BIN_PS -p $pid -o pid= 2>&1")['out']) != "";
}
j(['ok'=>!$alive1 || !$alive2, 'diag'=>$info, 'TERM'=>$term, 'alive_after_TERM'=>$alive1,
'KILL'=>$kill, 'alive_after_KILL'=>$alive2]);

```

```
}
```

```
}
```

j(['ok'=>true, 'diag'=>\$info, 'hint'=>'?pid=XXXX را بده. برای تست واقعی do=1 پارامتر kill اضافه کن.']);

توضیح :

```
session_start();
```

```
header('Content-Type: application/json; charset=utf-8');
```

سشن را آغاز می کند (برای شناخت کاربر/الگین).

نوع پاسخ را JSON با UTF-8 اعلام می کند.

```
function j($x){ echo json_encode($x, JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT);
exit; }
```

تابع کمکی برای خروج JSON زیبا (بدون escape یونیکد) و خاتمه اجرای اسکریپت.

```
function sh($cmd)

$ des = [1=>'[pipe','w'], 2=>'[pipe','w']];

$ p = proc_open($cmd, $des, $pipes);

$ out=""; $err=""; $rc=null;

if(is_resource($p))

$ out = stream_get_contents($pipes[1]); fclose($pipes[1]);

$ err = stream_get_contents($pipes[2]); fclose($pipes[2]);

$ rc = proc_close($p);

{
```

```
return ['rc'=>$rc,'out'=>$out,'err'=>$err,'cmd'=>$cmd];
{
```

تابع اجرای دستور شل با `proc_open`: خروجی استاندارد (`out`), خطأ (`err`), کُد بازگشت (`rc`) و خود دستور (`cmd`) را برمی‌گرداند.

```
function which($cands){ foreach($cands as $p){ if(@is_executable($p)) return $p; } return null; }
```

از میان مسیرهای پیشنهادی، اولین باینری قابل اجرا را پیدا می‌کند.

```
$pid = isset($_REQUEST['pid']) ? (int)$_REQUEST['pid'] : 0;
```

پارامتر `pid` را از GET/POST می‌خواند؛ اگر نبود، ..

```
$webuser = function_exists('posix_geteuid') && function_exists('posix_getpwuid')
? posix_getpwuid(posix_geteuid())['name'] ?? 'www-data'
: getenv('APACHE_RUN_USER') ?: (getenv('USER') ?: 'www-data');
```

نام کاربر وب‌서ور (فرایند PHP) را تعیین می‌کند؛ ترجیحاً از POSIX؛ در غیر این صورت از متغیرهای محیطی؛ پیش‌فرض .www-data

```
$BIN_PS = which(['/bin/ps','/usr/bin/ps']) ?: 'ps';
$BIN_KILL = which(['/bin/kill','/usr/bin/kill']);
$BIN_SUDO = which(['/usr/bin/sudo','/bin/sudo']);
```

مسیر اجرایی `ps`، `kill` و (در صورت وجود) `sudo` را پیدا می‌کند؛ اگر `ps` در مسیر ثابت نبود، از نام عمومی `ps` استفاده می‌کند.

```
$info] =
' session_uid' => $_SESSION['uid'] ?? null,
```

```

' web_user' => $webuser,
' paths] <=
' ps' => $BIN_PS,
' kill' => $BIN_KILL,
' sudo' => $BIN_SUDO,
,[

' sudo_nopass_check' => $BIN_SUDO ? sh("$BIN_SUDO -n true 2>&1") : null,
;[
```

شیء تشخیصی اولیه: شناسه سشن، نام کاربر وب، مسیر ابزارها، و نتیجه‌ی تست sudo -n true (اگر sudo هست: بررسی می‌کند بدون پسورد قابل اجراست یا نه).

```

if($pid > 0)

$ pUser = sh("$BIN_PS -o user= -p $pid 2>&1");
$ pComm = sh("$BIN_PS -o comm= -p $pid 2>&1");
$ info['process'] = ['pid'=>$pid, 'ps_user'=>$pUser, 'ps_comm'=>$pComm];
```

اگر pid معتبر است:

با ps صاحب پردازه (user) و نام فرمان (comm) را می‌خواند.

نتایج را در بخش process می‌گذارد.

بهی 1 فقط اگر پارامتر KILL و TERM // تست واقعی

```

if(isset($_REQUEST['do']) && $_REQUEST['do']=='1')

if(!$BIN_KILL){ j(['ok'=>false, 'error'=>'kill_not_found', 'diag'=>$info]); }
```

فقط اگر `do=1` ارسال شده باشد، تست واقعی ارسال سیگنال‌ها انجام می‌شود.

نبودن باینری `kill` → خروج با خطای تشخیصی.

```
$ term = sh(($BIN_SUDO ? "$BIN_SUDO -n " : ""))."${BIN_KILL}_TERM $pid 2>&1");
usleep(250000)
```

```
$ alive1 = trim(sh("${BIN_PS} -p $pid -o pid= 2>&1")['out']);" ==!
```

ارسال SIGTERM (مودبانه) با `kill -TERM` (در صورت وجود sudo با `n-`).

کمی صبر (ms250).

بررسی زنده‌بودن پردازه با `ps` (اگر خروجی PID خالی نباشد → زنده است).

```
$ kill = null; $alive2 = null;
```

```
if($alive1){
```

```
$ kill = sh(($BIN_SUDO ? "$BIN_SUDO -n " : ""))."${BIN_KILL}_KILL $pid 2>&1");
usleep(250000)
```

```
$ alive2 = trim(sh("${BIN_PS} -p $pid -o pid= 2>&1")['out']);" ==!
```

```
{
```

اگر بعد از TERM زنده بود، SIGKILL می‌فرستد؛ دوباره کمی صبر و چک زنده‌بودن.

```
j(['ok'=>!$alive1 || !$alive2, 'diag'=>$info, 'TERM'=>$term, 'alive_after_TERM'=>$alive1,
'KILL'=>$kill, 'alive_after_KILL'=>$alive2]);
```

```
{
```

{

خروج JSON نتیجه‌ی تست:

وقتی true می‌شود که پردازه پس از KILL یا TERM یا دیگر زنده نباشد.

همراه با جزئیات تشخیصی و خروجی‌های دستورات.

هم اضافه کن.`[.]; do=1` پارامتر kill را بده. برای تست واقعی `'pid=XXXX'` اگر pid داده نشده یا تست واقعی انجام نشده، خروج JSON تشخیصی عمومی + راهنمای نحوه‌ی فراخوانی.

#### خلاصه عملکرد

این اسکریپت یک اندپوینت تشخیصی کشن پردازه است: مسیر ابزارها و وضعیت sudo را گزارش می‌دهد؛ در صورت ارائه‌ی pid اطلاعات آن پردازه را می‌خواند؛ و اگر `do=1` بدهی، واقعاً SIGTERM و در صورت نیاز SIGKILL می‌فرستد، زنده‌بودن را بررسی می‌کند و همه‌ی نتایج را به صورت JSON برمی‌گرداند.

#### API

`blocklist..php:`

```
<?php

// api/blocklist.php

header('Content-Type: application/json; charset=utf-8');

$BLOCK_PATH = __DIR__ . '/../config/blocklist.json';

if (is_file($BLOCK_PATH)) {

 $j = file_get_contents($BLOCK_PATH);

 $arr = json_decode($j, true);

 if (!is_array($arr)) $arr = [];

 echo json_encode(['ok'=>true, 'blocked'=>$arr], JSON_UNESCAPED_UNICODE);
```

```
} else {
 echo json_encode(['ok'=>true, 'blocked'=>[]], JSON_UNESCAPED_UNICODE);
}
}
```

توصیح:

```
?>php
//api/blocklist.php
header('Content-Type: application/json; charset=utf-8');
```

فایل API برای گرفتن «بلاکلیست» (لیست رویدادهای مسدودشده).

خروجی را به شکل JSON با یونیکد تنظیم می‌کند.

```
$BLOCK_PATH = __DIR__ . '/../config/blocklist.json';
```

مسیر فایل JSON حاوی بلاکلیست را مشخص می‌کند.

این فایل در پوشه‌ی config (یک سطح بالاتر از api) قرار دارد.

```
if (is_file($BLOCK_PATH)) {
```

اگر فایل blocklist.json وجود داشته باشد، ادامه می‌دهد.

```
$ j = file_get_contents($BLOCK_PATH);
```

```
$ arr = json_decode($j, true);
```

```
;[] = if (!is_array($arr)) $arr
```

محتوای JSON فایل را می‌خواند.

آن را به آرایه‌ی PHP تبدیل می‌کند (true برای .(associative array

اگر decode موفق نبود یا چیزی غیر از آرایه برگشت، یک آرایه‌ی خالی جایگزین می‌کند.

```
echo json_encode(['ok'=>true, 'blocked'=>$arr], JSON_UNESCAPED_UNICODE);
```

خروجی JSON می‌سازد شامل:

→ نشانگر موفقیت ok

→ لیست بلاک شده‌ها (همان آرایه‌ی JSON)

با JSON\_UNESCAPED\_UNICODE متن فارسی و یونیکد خوانا می‌ماند.

{else}

```
echo json_encode(['ok'=>true, 'blocked'=>[], JSON_UNESCAPED_UNICODE);
{
```

اگر فایل اصلاً وجود نداشت، باز هم ok=true برمی‌گرداند، ولی blocked آرایه‌ی خالی است.

خلاصه عملکرد

این API وضعیت بلاک لیست را از فایل config/blocklist.json می‌خواند.

اگر فایل وجود داشت: محتوای آن (لیست برچسب‌ها یا آیتم‌های بلاک شده) به کلاینت داده می‌شود.

اگر نبود: آرایه‌ی خالی برمی‌گرداند.

خروجی همیشه ساختار ثابت دارد:

```
" }ok": true, "blocked{ [...] ":"
```

فایل JSON لوكال مدیریت می‌شود.

## API

cpu\_metrics\_list.php:

```
<?php
// api/cpu_metrics_list.php
header('Content-Type: application/json; charset=utf-8');
require_once __DIR__ . '/../config.php'; // برای STORAGE_PATH

// مسیر ذخیره تاریخچه (JSON Lines)
$HIST_FILE = STORAGE_PATH . '/metrics_history.jsonl';
$MAX_LINES = 200; // سقف نگهداری فایل

function read_cpu_stat(){
 $s = @file('/proc/stat');
 if(!$s) return null;
 foreach($s as $line){
 if(strpos($line, 'cpu ') === 0){
```

```

$parts = preg_split('/\s+/', trim($line));

return array_map('intval', array_slice($parts, 1, 10));

}

}

return null;

}

function cpu_percent(){

$a = read_cpu_stat();

if(!$a){ return null; }

usleep(200000); // 0.2s نمونه برداری

$b = read_cpu_stat();

if(!$b){ return null; }

$idleA = $a[3] + $a[4];

$idleB = $b[3] + $b[4];

$totalA = array_sum($a);

$totalB = array_sum($b);

$totald = $totalB - $totalA;

$idled = $idleB - $idleA;

if ($totald <= 0) return null;

$usage = (1.0 - ($idled / $totald)) * 100.0;

return max(0.0, min(100.0, $usage));

}

function ram_percent(){

$mi = @file('/proc/meminfo');

if(!$mi) return null;

$data = [];

```

```

foreach($mi as $ln){

 if (preg_match('/^(\w+):\s+(\d+)/', $ln, $m)){
 $data[$m[1]] = (int)$m[2]; // kB
 }
}

if(empty($data['MemTotal'])) return null;

$memTotal = $data['MemTotal'];

$memAvail = isset($data['MemAvailable']) ? $data['MemAvailable'] : ($data['MemFree'] ?? 0);

$used = max(0, $memTotal - $memAvail);

return ($used / $memTotal) * 100.0;
}

function cpu_temp_c(){

$cands = glob('/sys/class/thermal/thermal_zone*/temp');

if(!$cands) return null;

foreach($cands as $p){

 $raw = @file_get_contents($p);

 if($raw !== false){

 $v = (int)trim($raw);

 if($v > 1000) return $v / 1000.0;

 return floatval($v);
 }
}

return null;
}

function ip_addr(){

```

```

$out = @shell_exec('hostname -I');

if($out){

 $parts = preg_split('/\s+/', trim($out));

 foreach($parts as $ip){

 if (filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4) && $ip !== '127.0.0.1'){

 return $ip;

 }

 }

}

return $_SERVER['SERVER_ADDR'] ?? null;

}

function wifi_rssi(){

// اگر نشد RSSI تلاش برای گرفتن : null

$iw = @shell_exec('/sbin/iwconfig 2>/dev/null');

if(!$iw) $iw = @shell_exec('iwconfig 2>/dev/null');

if(!$iw) return null;

if (preg_match('/Signal level[=:]|\s*(-?\d+)\s*dBm/i', $iw, $m)) {

 return intval($m[1]);

}

// fallback /proc/net/wireless بسیار ساده از

$w = @file('/proc/net/wireless');

if($w && isset($w[2])){

 // فرمت: Inter-| sta| Quality | Discarded packets | Missed | WE

 // | link level noise ...

 // level از آن عبور می کنیم dBm است که ممکن است در Quality ستون سوم نباشد؛
}

```

```

}

return null;

}

// نمونه‌ی جدید بگیر

$now = date('Y-m-d H:i:s');

$cpu = cpu_percent();

$ram = ram_percent();

$temp = cpu_temp_c();

$ip = ip_addr();

$rssi = wifi_rssi();

$rec = [
 'created_at' => $now,
 'cpu_percent' => $cpu !== null ? round($cpu, 1) : null,
 'ram_percent' => $ram !== null ? round($ram, 1) : null,
 'cpu_temp_c' => $temp !== null ? round($temp, 1) : null,
 'ip_addr' => $ip,
 'wifi_rssi' => $rssi,
];

// نگه داریم (حداکثر 200 خط) تاریخچه را به صورت JSONL

if (!is_dir(dirname($HIST_FILE))) {
 @mkdir(dirname($HIST_FILE), 0755, true);
}

```

```

$ok = @file_put_contents($HIST_FILE, json_encode($rec,
JSON_UNESCAPED_UNICODE)."\\n", FILE_APPEND | LOCK_EX);

// خواندن آخرین 20 رکورد (از انتهای فایل)
$rows = [];

if (is_file($HIST_FILE)) {

 // بهینه ساده tail
 $lines = @file($HIST_FILE, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);

 if($lines !== false){

 $lines = array_slice($lines, -20);

 $lines = array_reverse($lines); // جدیدترین در بالا

 foreach($lines as $ln){

 $j = json_decode($ln, true);

 if (is_array($j)) $rows[] = $j;

 }
 }
}

```

```

// نگهداری سقف خطوط (فقط گاهی اوقات برای سبکسازی انجام بدھیم)

if (is_file($HIST_FILE)) {

 $lines = @file($HIST_FILE, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);

 if ($lines !== false && count($lines) > $MAX_LINES){

 // فقط آخرین $MAX_LINES خط را نگه داریم

 $last = array_slice($lines, -$MAX_LINES);

 @file_put_contents($HIST_FILE, implode("\\n", $last)."\\n", LOCK_EX);
 }
}

```

```
}
```

```
echo json_encode(['rows' => $rows], JSON_UNESCAPED_UNICODE);
```

توضیح :

```
//api/cpu_metrics_list.php
```

```
header('Content-Type: application/json; charset=utf-8');
```

اعلام می کند خروجی API به صورت JSON با UTF-8 است.

```
require_once __DIR__ . '/../config.php // برای 'STORAGE_PATH
```

فایل تنظیمات را می آورد تا به ثابت هایی مثل STORAGE\_PATH دسترسی داشته باشد.

```
$HIST_FILE = STORAGE_PATH . '/metrics_history.jsonl';
```

// سقف نگهداری فایل MAX\_LINES = 200\$

مسیر فایل تاریخچه‌ی متريکها (به صورت JSON Lines) و حداکثر تعداد خطوط نگهداری شده را تعیين می کند.

```
function read_cpu_stat() {
 $s = @file('/proc/stat');
 if(!$s) return null;
 foreach($s as $line) {
 if(strpos($line, 'cpu ') === 0) {
 $parts = preg_split('/\s+/', trim($line));

```

```

 return array_map('intval', array_slice($parts, 1, 10));

{
{
return null;
{

```

تابع کمکی برای خواندن ردیف کلی CPU از proc/stat/ (مقادیر زمانی حالت‌های مختلف CPU). آرایه‌ای از اعداد صحیح برمی‌گرداند.

```

function cpu_percent}()

$ a = read_cpu_stat;()

if(!$a){ return null; }

 usleep(200000); // 0.2s نمونه برداری

$ b = read_cpu_stat;()

if(!$b){ return null; }

$ idleA = $a[3] + $a[4]

$ idleB = $b[3] + $b[4]

$ totalA = array_sum($a);

$ totalB = array_sum($b);

$ totald = $totalB - $totalA;

$ idled = $idleB - $idleA;

if ($totald <= 0) return null;

$ usage = (1.0 - ($idled / $totald)) * 100.0;

return max(0.0, min(100.0, $usage));

{

```

- با دو بار نمونه برداری از /proc/stat و اختلاف گیری کل/idle در فاصله‌ی ۰.۲ ثانیه، درصد استفاده‌ی CPU را محاسبه و بین تا ۱۰۰ کمپ می‌کند.

```
function ram_percent{}()
$ mi = @file('/proc/meminfo');
if(!$mi) return null;
$ data=[];
foreach($mi as $ln){
 if(preg_match('/^(\w+):\s+(\d+)/', $ln, $m)){
 $ data[$m[1]] = (int)$m[2]; // kB
 }
}
if(empty($data['MemTotal'])) return null;
$ memTotal = $data['MemTotal'];
$ memAvail = isset($data['MemAvailable']) ? $data['MemAvailable'] : ($data['MemFree'] ?? 0);
$ used = max(0, $memTotal - $memAvail);
return ($used / $memTotal) * 100.0;
}

از /proc/meminfo کل حافظه و مقدار «در دسترس» را می‌خواند، استفاده شده را حساب می‌کند و درصد مصرف RAM را برمی‌گرداند.
```

```
function cpu_temp_c{}()
$cands = glob('/sys/class/thermal/thermal_zone*/temp');
if(!$cands) return null;
```

```

foreach($cands as $p)
{
$ raw = @file_get_contents($p);
if($raw !== false)
{
$ v = (int)trim($raw);
if($v > 1000) return $v / 1000.0;
return floatval($v);
}
{
return null;
}

```

از فایل‌های thermal در sysfs دمای CPU را می‌خواند. اگر مقدار در میلی‌درجه باشد (بزرگ‌تر از ۱۰۰۰)، به درجه سانتی‌گراد تقسیم بر ۱۰۰۰ می‌کند.

```

function ip_addr }()
$ out = @shell_exec('hostname -I');
if($out)
{
$ parts = preg_split('/\s+/', trim($out));
foreach($parts as $ip)
{
if(filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4) && $ip !== '127.0.0.1')}
return $ip;
}
{
return $_SERVER['SERVER_ADDR'] ?? null;

```

{

با IP - I ها را می‌گیرد و اولین IPv4 غیر localhost را بر می‌گرداند؛ در غیر این صورت از آدرس سرور در درخواست وب استفاده می‌کند.

```
function wifi_rssi(){}
```

RSSI: اگر نشد null // تلاش برای گرفتن

```
$ iw = @shell_exec('/sbin/iwconfig 2>/dev/null');
if(!$iw) $iw = @shell_exec('iwconfig 2>/dev/null');
if(!$iw) return null;
if(preg_match('/Signal level[=:]\s*(-?\d+)\s*dBm/i', $iw, $m)) {
 return intval($m[1]);
}
```

{

// fallback/ بسیار ساده از proc/net/wireless

```
$ w = @file('/proc/net/wireless');
if($w && isset($w[2])) {
```

نیست؛ اینجا صرفاً رد می‌شود. dBm // توضیح: فرمت این فایل استاندارد

{

```
return null;
```

{

قدرت سیگنال وای‌فای (RSSI) را از خروجی iwconfig به dBm استخراج می‌کند؛ اگر نشد، تلاش سبک از proc/net/wireless/ (اما عملاً null بر می‌گرداند).

```
$now = date('Y-m-d H:i:s');
```

```

$cpu = cpu_percent;()
$ram = ram_percent;()
$temp = cpu_temp_c;()
$ip = ip_addr;()
$rss = wifi_rssi;()

```

نمونه‌ی جدید متریک‌ها را همین‌الآن جمع‌آوری می‌کند.

```

$rec] =
' created_at' => $now,
' cpu_percent'=> $cpu !== null ? round($cpu,1) : null,
' ram_percent'=> $ram !== null ? round($ram,1) : null,
' cpu_temp_c' => $temp !== null ? round($temp,1) : null,
' ip_addr' => $ip,
' wifi_rssi' => $rss,
;[

```

رکورد متریک‌ها را با گرد کردن یک رقم اعشار برای CPU/RAM/Temp می‌سازد.

```

if (!is_dir(dirname($HIST_FILE)))}

@ mkdir(dirname($HIST_FILE), 0755, true);

{
$ok = @file_put_contents($HIST_FILE, json_encode($rec,
JSON_UNESCAPED_UNICODE)."\\n", FILE_APPEND | LOCK_EX);

```

اگر پوشه‌ی تاریخچه نبود می‌سازد؛ سپس رکورد فعلی را به صورت یک خط JSON در انتهای فایل JSON اضافه می‌کند (با قفل فایل).

```

$rows[] =
if (is_file($HIST_FILE)) {
$ lines = @file($HIST_FILE, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
if($lines !== false) {
$ lines = array_slice($lines, -20);
$ lines = array_reverse($lines) // ;
foreach($lines as $ln) {
$ j = json_decode($ln, true);
if (is_array($j)) $rows[] = $j;
}
}
}

```

برای پاسخ، آخرین ۲۰ رکورد را از انتهای فایل می‌خواند، ترتیب‌شان را برعکس می‌کند تا جدیدترین‌ها اول باشند، و JSON هر خط را به آرایه اضافه می‌کند.

```

if (is_file($HIST_FILE)) {
$ lines = @file($HIST_FILE, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
if ($lines !== false && count($lines) > $MAX_LINES) {
$ last = array_slice($lines, -$MAX_LINES);
@ file_put_contents($HIST_FILE, implode("\n", $last)."\n", LOCK_EX);
}
}

```

اگر تعداد خطوط فایل از سقف تعیین شده (۲۰۰) بیشتر شد، فقط آخرین ۲۰۰ خط را نگه می‌دارد تا فایل بزرگ نشود.

```
echo json_encode(['rows' => $rows], JSON_UNESCAPED_UNICODE);
```

پاسخ API آرایه‌ای از آخرین متریک‌ها در فیلد rows به صورت JSON

خلاصه‌ی عملکرد

این API متریک‌های لحظه‌ای RAM%, CPU%، دمای IP و RSSI را از سیستم می‌خواند، یک رکورد را به فایل JSON تاریخچه اضافه می‌کند، سپس آخرین ۲۰ رکورد را خوانده و به صورت JSON برمی‌گرداند؛ در عین حال حجم فایل تاریخچه را حداکثر روی ۲۰۰ خط نگه می‌دارد.

## Assets

### Style.css:

```
html, body { direction: rtl; }
```

```
body { background-color: #0b1320; color: #e6e9ef; }
```

```
.bg-pattern {
```

```
background: radial-gradient(80rem 30rem at 10% -10%, #1b2a4a, transparent),
```

```
radial-gradient(70rem 20rem at 110% -10%, #192338, transparent),
```

```
linear-gradient(180deg, #0b1320, #0b1320);
```

```
}
```

```
.card-glass {
```

```
background: rgba(255,255,255,0.06);
```

```
border: 1px solid rgba(255,255,255,0.1);
```

```
backdrop-filter: blur(6px);
```

```
border-radius: 1rem;
```

```
}
```

```
.navbar-blur {
 backdrop-filter: blur(6px);
 background: rgba(255,255,255,0.06);
 border-bottom: 1px solid rgba(255,255,255,0.1);
}

.navbar .nav-link, .navbar .navbar-brand { color: #e6e9ef; }
.navbar .nav-link:hover { color: #9cc2ff; }

.kpi {
 background: linear-gradient(180deg, rgba(32,52,89,0.45), rgba(22,34,61,0.5));
 border: 1px solid rgba(255,255,255,0.08);
 border-radius: .9rem;
}

.kpi-label { font-size: .9rem; color: #c0c7d1; }
.kpi-value { font-size: 2rem; font-weight: 800; letter-spacing: .5px; }
.kpi-sub { font-family: monospace; font-size: .9rem; color: #c0c7d1; }

.table { color: #e6e9ef; }
.table thead, .table-light { color: #101522; background: #e8eefb; }

.card, .table, .form-control, .btn { border-radius: .9rem; }
.form-control, .btn {
 border: 1px solid rgba(255,255,255,0.15);
```

```

background-color: rgba(255,255,255,0.04);

color: #e6e9ef;

}

.form-control::placeholder { color: #9fb2d0; }

.btn-primary { background-color: #357ae8; border-color: #357ae8; }

.btn-primary:hover { background-color: #2d6bd0; border-color: #2d6bd0; }

.table-hover tbody tr:hover { background-color: rgba(255,255,255,0.05); }

.alert-danger {

background: rgba(255, 107, 107, .15);

border-color: rgba(255, 107, 107, .35);

color: #ff8a8a;

}

.kpi-value.good { color: #4caf50; } /* سبز */

.kpi-value.warn { color: #ffc107; } /* زرد */

.kpi-value.bad { color: #f44336; } /* قرمز */

/* ===== */

/* دکمه‌های ری‌استارت و خاموش */

/* ===== */

.card .btn-warning,
.card .btn-danger {

```

```
background-color: #000 !important; /*پس زمینه مشکی*/
color: #fff !important; /*متن سفید*/
border: 1px solid #222 !important; /*کادر تیره*/
border-radius: .6rem !important; /*گردی کمتر*/
padding: .35rem 1rem !important; /*ارتفاع و عرض متعادل*/
font-size: .9rem !important;
font-weight: 600 !important;
transition: background-color 0.2s ease-in-out;
}
```

```
/* حالت hover */
.card .btn-warning:hover,
.card .btn-danger:hover {
background-color: #222 !important;
color: #fff !important;
}
```

```
/* (کلیک) → قرمز حالت active */
.card .btn-warning:active,
.card .btn-danger:active {
background-color: #d32f2f !important;
border-color: #d32f2f !important;
color: #fff !important;
}
```

```
/* حالت disabled */

.card .btn-warning:disabled,
.card .btn-danger:disabled {
background-color: #000 !important;
color: #aaa !important;
opacity: 0.6 !important;
}
```

```
/* متن راهنمای کنار دکمه‌ها */

.card .text-muted {
color: #fff !important; /* سفید */
opacity: 0.9 !important;
font-size: .85rem !important;
}

.login-title {
color: #fffff !important; /* سفید */
font-size: 2rem; /* اندازه بزرگ‌تر */
font-weight: 700; /* ضخیم‌تر */
font-family: "Vazirmatn", "Tahoma", sans-serif; /* فونت فارسی/لاتین زیبا */
letter-spacing: .5px; /* فاصله‌ی بین حروف کمی بیشتر */
}

#evPerf {
color: #fff !important; /* سفید */
font-weight: 500; /* کمی پرنگ‌تر */
```

```

}

/* وضعیت پایین جدول رو سفید کن */

.ev-status,
.ev-status span,
.ev-status #evPerf {
 color: #fff !important;
 font-weight: 400;
}

/* دکمه‌های داخل کارت روی موبایل تمام‌عرض و خوش‌خوان */

.card .btn { width: 100%; }

/* روی موبایل کمی جمع‌وجورتر KPI تیتر/مقادیر */

@media (max-width: 576px) {

 .kpi .kpi-value { font-size: clamp(14px, 4.2vw, 22px); }

 .kpi .kpi-label, .kpi .kpi-sub { font-size: 12px; }

 .card .card-body { padding: 12px 14px; }

 .card-desc { font-size: .9rem; }

}

```

توضیح :

Assets

app.js:

```

// assets/app.js

const IOT = () => {
 const fmt = d => new Date(d).toLocaleString('fa-IR');

 // ===== Dashboard KPIs (real) =====
 async function pollHealth() {
 try {
 const r = await fetch('api/system_health.php', { cache: 'no-store' });
 if (!r.ok) return;
 const j = await r.json();

 function colorize(id, val) {
 const el = document.getElementById(id);
 if (!el) return;
 el.className = 'kpi-value';
 let cls = 'good';
 if (val >= 80) cls = 'bad';
 else if (val >= 50) cls = 'warn';
 el.classList.add(cls);
 el.textContent = (id === 'temp')
 ? `${Number(val).toFixed(1)}°C`
 : `${Number(val).toFixed(1)}%`;
 }

 colorize('cpu', j.cpu_percent ?? 0);
 colorize('ram', j.ram_percent ?? 0);
 }
 }
}

```

```

colorize('temp', j.cpu_temp_c ?? 0);

const netEl = document.getElementById('net');

if (netEl) {
 const ip = j.ip_addr ?? '—';
 const rssi = (j.wifi_rssi ?? '—') + (j.wifi_rssi !== null && j.wifi_rssi !== undefined ? ' dBm' : '');
 netEl.replaceChildren(document.createTextNode(` ${ip} / RSSI ${rssi}`));
}

} catch (e) { /* ignore */ }

}

// ===== Events (page) with pagination + filters =====

let EV_STATE = { page: 1, per: 50 };

let EV_TIMER = null;

let EV_BUSY = false;

function applyEventFilters() {
 EV_STATE.page = 1;
 refreshEvents(false);
}

async function refreshEvents(first) {
 if (EV_BUSY) return;
 EV_BUSY = true;
 const btn = document.querySelector('button.btn.btn-sm.btn-primary');

```

```

const evPerfEl = document.getElementById('evPerf');

شروع تایمر کلاینت // performance.now()

try {
 قفل عرض دکمه برای جلوگیری از پرش //

 if (btn) {
 btn.classList.add('btn-filter-fixed'); // در CSS: min-width

 const w = btn.getBoundingClientRect().width;
 btn.style.width = Math.ceil(w) + 'px';

 btn.disabled = true;

 btn.dataset._orig = btn.innerHTML;
 btn.innerHTML = ' اعمال فیلتر';
 }

 if (evPerfEl) { evPerfEl.textContent = 'در حال جستجو...'; }
}

const params = new URLSearchParams();

const s = document.getElementById('start')?.value ?? '';
const e = document.getElementById('end')?.value ?? '';
const q = (document.getElementById('search')?.value ?? '').trim();
const confMin = document.getElementById('confMin')?.value ?? '';
const confMax = document.getElementById('confMax')?.value ?? '';

const perSel = document.getElementById('evPer');

if (perSel) {

```

```

const perVal = parseInt(perSel.value || '50', 10);
if (!isNaN(perVal)) EV_STATE.per = perVal;
}

if (q) params.set('q', q);
if (s) params.set('start', s);
if (e) params.set('end', e);
if (confMin !== "") params.set('conf_min', confMin);
if (confMax !== "") params.set('conf_max', confMax);
params.set('page', EV_STATE.page);
params.set('per', EV_STATE.per);

const r = await fetch('api/events_list.php?' + params.toString(), { cache: 'no-store' });
const txt = await r.text();
let j; try { j = JSON.parse(txt); } catch { j = null; }
if (!r.ok || !j || j.ok === false) {
 console.error('events_list error:', txt);
 alert('خطا در واکشی رویدادها. لطفاً صفحه را رفرش کنید.' + (j?.message ? ('\n' + j.message) : ""));
 return;
}

const tb = document.getElementById('eventsFull'); if (!tb) return;
tb.innerHTML = "";
(j.rows || []).forEach((x, i) => {
 const tr = document.createElement('tr');
 tr.innerHTML =

```

```

` <td>${(EV_STATE.page - 1) * EV_STATE.per + i + 1}</td>
<td>${fmt(x.ts)}</td>
<td>${x.label}</td>
<td>${Number(x.confidence).toFixed(1)}%</td>
<td>${x.node_id}</td>`;
tb.appendChild(tr);
});

EV_STATE.page = j.page ?? EV_STATE.page;
EV_STATE.per = j.per ?? EV_STATE.per;
const pages = j.pages ?? 1;
const total = j.total ?? 0;

const elPage = document.getElementById('evPage');
const elPages = document.getElementById('evPages');
const elTotal = document.getElementById('evTotal');
if (elPage) elPage.textContent = EV_STATE.page;
if (elPages) elPages.textContent = pages;
if (elTotal) elTotal.textContent = total;

const prevBtn = document.getElementById('evPrev');
const nextBtn = document.getElementById('evNext');
if (prevBtn) {
 prevBtn.disabled = (EV_STATE.page <= 1);
 prevBtn.onclick = () => { if (EV_STATE.page > 1) { EV_STATE.page--; refreshEvents(false); } };
}

```

```

 }

if (nextBtn) {

 nextBtn.disabled = (EV_STATE.page >= pages);

 nextBtn.onclick = () => { if (EV_STATE.page < pages) { EV_STATE.page++; refreshEvents(false); } };

}

```

از صفحه ۱ per تغییر //

```

if (perSel && !perSel._bound) {

 perSel.addEventListener('change', () => { EV_STATE.page = 1; refreshEvents(false); });

 perSel._bound = true;

}

```

صفحه ۱: فیلترها // change/Enter →

```

const startEl = document.getElementById('start');

const endEl = document.getElementById('end');

const qEl = document.getElementById('search');

const cminEl = document.getElementById('confMin');

const cmaxEl = document.getElementById('confMax');

[startEl, endEl, qEl, cminEl, cmaxEl].forEach(el => {

 if (el && !el._bound) {

 el.addEventListener('change', () => { EV_STATE.page = 1; refreshEvents(false); });

 if (el.tagName === 'INPUT') {

 el.addEventListener('keyup', (ev) => { if (ev.key === 'Enter') { EV_STATE.page = 1; refreshEvents(false); } });

 }
 }
})

```

```

el._bound = true;
}

});

// 🌿 نمایش زمان جستجو/اتمام

const t1 = performance.now();

const clientMs = Math.max(0, t1 - t0);

const hasDb = (typeof j.bench_ms === 'number');

const dbStr = hasDb ? `${j.bench_ms.toFixed(0)} ms` : '—';

const totalStr= `${clientMs.toFixed(0)} ms`;

let indexStr = ":";

if (j.explain && j.explain[0]) {

 const k = j.explain[0].key || j.explain[0].partitions || null;

 if (k) indexStr = ` · index=${k}`;

}

if (evPerfEl) {

 evPerfEl.textContent = ` کل = ${totalStr} ${indexStr}`;

}

if (first) {

 if (EV_TIMER) clearInterval(EV_TIMER);

 EV_TIMER = setInterval(() => refreshEvents(false), 5000);

}

} catch (e) {

 console.error(e);
}

```

```

 alert('خطای غیرمنتظره در بارگذاری رویدادها');

 } finally {
 EV_BUSY = false;
 if (btn) {
 btn.disabled = false;
 btn.innerHTML = (btn.dataset._orig ?? '<i class="bi bi-funnel"></i>');
 // کمی تأخیر تا متن جا بیفت؛ سپس قفل عرض برداشته شود تا پرس نکند
 setTimeout(() => { btn.style.width = ""; }, 50);
 }
 }
}

// ===== Processes (page) + Kill =====
async function refreshProcesses() {
 try {
 const r = await fetch('api/processes_list.php', { cache: 'no-store' });
 if (!r.ok) return;
 const j = await r.json();

 const tb = document.getElementById('procFull'); if (!tb) return;
 tb.innerHTML = "";

 (j.rows || []).forEach((x, i) => {
 const tr = document.createElement('tr');
 const btn =

```

```

` <button class="btn btn-sm btn-danger" onclick="IOT.killPid(${x.pid}, '${x.user}')">
 <i class="bi bi-x-circle"></i>
</button> `;

tr.innerHTML =
` <td>${i + 1}</td>
<td>${x.pid}</td>
<td>${x.user ?? '—'}</td>
<td>${x.name}</td>
<td>${x.cpu_percent}</td>
<td>${x.mem_percent}</td>
<td>${fmt(x.created_at)}</td>
<td>${btn}</td> `;

tb.appendChild(tr);

});

} catch (e) { /* ignore */ }

}

```

```

async function killPid(pid, user) {
 if (!confirm(`مطمئن هستید؟ ${pid} (${user}) را از خاتمه پردازه`)) return;

 try {
 const form = new FormData();
 form.append('pid', pid);

 const r = await fetch('api/process_kill.php', { method: 'POST', body: form, cache: 'no-store' });
 }
}

```

```

const txt = await r.text();

let j; try { j = JSON.parse(txt); } catch { j = null; } // ← txt رفع باگ شود درست پارس شود

if (!r.ok || !j || !j.ok) {

 const msg = j?.message || j?.error || txt || 'نامشخص';

 alert(`خطا در Kill:\n${msg}`);

 if (j?.stderr) console.error('stderr:', j.stderr);

 return;
}

alert(`پردازه ${pid} یافت خاتمه.`);
refreshProcesses();

} catch (e) {

 alert(`پردازه ${pid} یافت خاتمه.`);
}

// ===== Metrics (page) =====

async function loadMetrics() {

 try {

 const r = await fetch('api/cpu_metrics_list.php', { cache: 'no-store' });

 if (!r.ok) return;

 const j = await r.json();

 const tb = document.getElementById('metricsT'); if (!tb) return;

 const st = document.getElementById('st'),

```

```

ip = document.getElementById('ip'),
rss = document.getElementById('rss');

tb.innerHTML = "";

(j.rows || []).forEach(x => {
 const tr = document.createElement('tr');
 tr.innerHTML =
 `<td>${fmt(x.created_at)}</td>
<td>${x.cpu_percent}</td>
<td>${x.ram_percent}</td>
<td>${x.cpu_temp_c ?? '—'}</td>
<td>${x.ip_addr ?? '—'}</td>
<td>${x.wifi_rssi ?? '—'}</td>`;
 tb.appendChild(tr);
});

if (j.rows?.length) {
 const last = j.rows[0];
 if (st) st.textContent = 'آنلاین';
 if (ip) ip.textContent = last.ip_addr ?? '—';
 if (rss) rss.textContent = (last.wifi_rssi ?? '—') + (last.wifi_rssi ? ' dBm' : '');
} else {
 if (st) st.textContent = '—';
 if (ip) ip.textContent = '—';
 if (rss) rss.textContent = '—';
}

```

```

} catch (e) { /* ignore */ }

}

// ===== System Specs (page) =====

async function loadSpecs() {

try {

const r = await fetch('api/system_specs.php', { cache: 'no-store' });

if (!r.ok) return;

const j = await r.json();

const set = (id, val) => { const el = document.getElementById(id); if (el) el.textContent =
(val ?? '—'); };

// مشخصات پایه

set('dev', j.device_model);

set('host', j.hostname);

set('os', j.os);

set('kernel', j.kernel);

set('arch', j.arch);

set('cpu', j.cpu_model);

set('cores', j.cpu_cores);

// RAM (table + progress)

set('ram_total', j.mem_total_mb != null ? `${j.mem_total_mb} MB` : '—');

set('ram_used', j.mem_used_mb != null ? `${j.mem_used_mb} MB` : '—');

set('ram_free', j.mem_free_mb != null ? `${j.mem_free_mb} MB` : '—');

```

```

const ramBar = document.getElementById('ram_bar');

if (ramBar && j.mem_total_mb && j.mem_used_mb != null) {
 const p = Math.max(0, Math.min(100, (j.mem_used_mb * 100) / j.mem_total_mb));
 ramBar.style.width = p.toFixed(0) + '%';
 ramBar.textContent = p.toFixed(0) + '%';
 ramBar.classList.remove('bg-success', 'bg-warning', 'bg-danger');
 if (p >= 80) ramBar.classList.add('bg-danger');
 else if (p >= 50) ramBar.classList.add('bg-warning');
 else ramBar.classList.add('bg-success');
}

// Disk (table + progress)

set('disk_total', j.disk_total_mb != null ? ` ${j.disk_total_mb} MB` : '—');
set('disk_used', j.disk_used_mb != null ? ` ${j.disk_used_mb} MB` : '—');
set('disk_free', j.disk_free_mb != null ? ` ${j.disk_free_mb} MB` : '—');

const diskBar = document.getElementById('disk_bar');

if (diskBar && j.disk_total_mb && j.disk_used_mb != null) {
 const p = Math.max(0, Math.min(100, (j.disk_used_mb * 100) / j.disk_total_mb));
 diskBar.style.width = p.toFixed(0) + '%';
 diskBar.textContent = p.toFixed(0) + '%';
}

} catch (e) { /* ignore */ }

}

// ===== KPI mini on system page =====

async function pollHealthKPI(prefix = "") {

```

```

try {

 const r = await fetch('api/system_health.php', { cache: 'no-store' });
 if (!r.ok) return;
 const j = await r.json();

 function colorizeEl(el, val, isTemp = false) {
 if (!el) return;
 el.className = 'kpi-value';
 let cls = 'good';
 if (!isTemp) {
 if (val >= 80) cls = 'bad';
 else if (val >= 50) cls = 'warn';
 el.classList.add(cls);
 el.textContent = isNaN(val) ? '—' : `${Number(val).toFixed(1)}%`;
 } else {
 if (val >= 75) cls = 'bad';
 else if (val >= 60) cls = 'warn';
 el.classList.add(cls);
 el.textContent = isNaN(val) ? '—' : `${Number(val).toFixed(1)}°C`;
 }
 }

 colorizeEl(document.getElementById(prefix + 'cpuK'), j.cpu_percent ?? NaN, false);
 colorizeEl(document.getElementById(prefix + 'ramK'), j.ram_percent ?? NaN, false);
 colorizeEl(document.getElementById(prefix + 'tmpK'), j.cpu_temp_c ?? NaN, true);
}

```

```

const ipEl = document.getElementById(prefix + 'ipK');

if (ipEl) ipEl.textContent = 'IP: ' + (j.ip_addr ?? '—');

const rs = document.getElementById(prefix + 'rssI');

if (rs) rs.textContent = 'RSSI: ' + ((j.wifi_rssi ?? '—') + (j.wifi_rssi ? ' dBm' : ""));

} catch (e) { /* ignore */ }

}

// ===== Power controls (reboot/shutdown) =====

async function sysPower(action) {

if (!['reboot', 'shutdown'].includes(action)) return;

const fa = action === 'reboot' ? 'ریاستارت' : 'خاموش';

if (!confirm(`سیستم مطمئن هستید؟ آیا از ${fa} باز`)) return;

const btn = typeof event !== 'undefined' ? event.currentTarget : null;

try {

if (btn) { btn.disabled = true; btn.dataset._orig = btn.innerHTML; btn.innerHTML = ' در حال ارسال...'; }

const fd = new FormData();

fd.append('action', action);

const r = await fetch('api/system_power.php', { method: 'POST', body: fd, cache: 'no-store' });

const txt = await r.text();

let j; try { j = JSON.parse(txt); } catch { j = null; }

if (!r.ok || !j?.ok) {

```

```

 alert('خطا: ' + (j?.error || txt || 'نامشخص'));
 } else {
 alert(`رسال شد دستور ${fa}`);
 }
} catch (e) {
 alert('خطای شبکه/سرور');
} finally{
 if (btn) { btn.disabled = false; btn.innerHTML = btn.dataset._orig || btn.innerHTML; }
}
}

// ===== Init bindings =====

function initDashboard() { pollHealth(); setInterval(pollHealth, 3000); }

function initEventsPage() { refreshEvents(true); }

function initProcessesPage() { refreshProcesses(); setInterval(refreshProcesses, 5000); }

function initMetricsPage() { loadMetrics(); setInterval(loadMetrics, 5000); }

function initSystemPage() { loadSpecs(); pollHealthKPI(""); setInterval(() =>
pollHealthKPI(""), 3000); }

return {
 initDashboard,
 initEventsPage,
 initProcessesPage,
 initMetricsPage,
 initSystemPage,
 refreshEvents,
}

```

```
applyEventFilters,
killPid,
sysPower
};
}());
```

توضیح :

### تنظیمات کلی

```
html, body { direction: rtl; }
body { background-color: #0b1320; color: #e6e9ef; }
```

- کل صفحه راستبهچپ (فارسی) است.

پس زمینه تیره (#0b1320) و متن خاکستری روشن.

### الگوهای پس زمینه

```
.bg-pattern {
background: radial-gradient(...), radial-gradient(...), linear-gradient(...);
}
```

- ترکیب چند گرادیان برای یک پس زمینه مدرن و گرافیکی.

### شیشه‌ای کردن کارت‌ها و منو

```
.card-glass { background: rgba(255,255,255,0.06); border: 1px solid rgba(...); backdrop-
filter: blur(6px); border-radius: 1rem; }
.navbar-blur { backdrop-filter: blur(6px); background: rgba(255,255,255,0.06); border-
bottom: 1px solid rgba(...); }
```

- کارت‌ها و نوار بالای صفحه (navbar) نیمه شفاف و مات گونه می‌شوند (افکت شیشه‌ای).

## استایل نوار بالا (navbar)

```
.navbar .nav-link, .navbar .navbar-brand { color: #e6e9ef; }
```

```
.navbar .nav-link:hover { color: #9cc2ff; }
```

- متن منو و برنده سفید.

- هنگام `hover` آبی روشن.

## استایل KPI کارت شاخص‌ها

```
.kpi { background: linear-gradient(...); border: 1px solid rgba(...); border-radius: .9rem; }
```

```
.kpi-label { font-size: .9rem; color: #c0c7d1; }
```

```
.kpi-value { font-size: 2rem; font-weight: 800; letter-spacing: .5px; }
```

```
.kpi-sub { font-family: monospace; font-size: .9rem; color: #c0c7d1; }
```

- هر KPI یک کارت کوچک با گرادیان و کادر کمرنگ دارد.

- مقدار KPI بزرگ و پرنگ نمایش داده می‌شود.

- متن توضیح (sub) با فونت مونو.

## جدول و فرم‌ها

```
.table { color: #e6e9ef; }
```

```
.table thead, .table-light { color: #101522; background: #e8eefb; }
```

```
.card, .table, .form-control, .btn { border-radius: .9rem; }
```

```
.form-control, .btn { border: 1px solid rgba(...); background-color: rgba(...); color: #e6e9ef; }
```

- جدول‌ها متن روشن دارند ولی سرستون‌ها پس زمینه روشن با متن تیره.

- همه‌ی اجرا (کارت، جدول، دکمه‌ها) گوشش‌های گرد دارند.

- ورودی‌ها و دکمه‌ها پس زمینه نیمه‌شفاف و متن سفید دارند.

## دکمه‌ها

```
.btn-primary { background-color: #357ae8; border-color: #357ae8; }
```

```
.btn-primary:hover { background-color: #2d6bd0; border-color: #2d6bd0; }
```

- دکمه اصلی آبی، در حالت **hover** کمی تیره‌تر.

## جدول **hover**

```
.table-hover tbody tr:hover { background-color: rgba(255,255,255,0.05); }
```

- ردیف جدول هنگام هاور کمی روشن‌تر می‌شود.

## پیام هشدار

```
.alert-danger { background: rgba(255, 107, 107, .15); border-color: rgba(...); color: #ff8a8a; }
```

- آلام قرمز نیمه‌شفاف با متن قرمز روشن.

## رنگ **KPI** بر اساس وضعیت

```
.kpi-value.good { color: #4caf50; } /* سبز */
```

```
.kpi-value.warn { color: #ffc107; } /* زرد */
```

```
.kpi-value.bad { color: #f44336; } /* قرمز */
```

- بر اساس وضعیت (خوب، هشدار، بد) رنگ مقدار **KPI** تغییر می‌کند.

## دکمه‌های ریاستارت و خاموش

```
.card .btn-warning, .card .btn-danger {
```

```
background-color: #000 !important; color: #fff !important;
```

```
border: 1px solid #222 !important; border-radius: .6rem !important;
```

```
padding: .35rem 1rem !important; font-size: .9rem !important; font-weight: 600 !important;
```

```
transition: background-color 0.2s ease-in-out;
```

```
}
```

```
.card .btn-warning:hover, .card .btn-danger:hover { background-color: #222 !important; }
```

```
.card .btn-warning:active, .card .btn-danger:active { background-color: #d32f2f !important; border-color: #d32f2f !important; }
```

```
.card .btn-warning:disabled, .card .btn-danger:disabled { background-color: #000 !important; color: #aaa !important; opacity: 0.6 !important; }
```

- دکمه‌های داخل کارت برای خاموش/اری استارت مشکی، با `hover` خاکستری، و هنگام `active` قرمز.
- دکمه‌های خاکستری کمرنگ، `disabled` → متن راهنمایی

```
.card .text-muted { color: #fff !important; opacity: 0.9 !important; font-size: .85rem !important; }
```

- متن‌های راهنمایی سفید و خواناتر.

## عنوان لاغین

```
.login-title { color: #fff !important; font-size: 2rem; font-weight: 700; font-family: "Vazirmatn", "Tahoma", sans-serif; letter-spacing: .5px; }
```

- عنوان صفحه‌ی لاغین بزرگ، ضخیم، سفید، و با فونت فارسی/لاتین زیبا.

## وضعیت جدول پایین

```
#evPerf { color: #fff !important; font-weight: 500; }
```

```
.ev-status, .ev-status span, .ev-status #evPerf { color: #fff !important; font-weight: 400; }
```

- بخش وضعیت و شمارش رکوردها در پایین جدول سفید نمایش داده می‌شود.

## حالت موبایل

```
.card .btn { width: 100%; }
```

```
@media (max-width: 576px) {
```

```
 .kpi .kpi-value { font-size: clamp(14px, 4.2vw, 22px); }
```

```
 .kpi .kpi-label, .kpi .kpi-sub { font-size: 12px; }
```

```
 .card .card-body { padding: 12px 14px; }
```

```
 .card-desc { font-size: .9rem; }
```

```
}
```

- دکمه‌های کارت‌ها در موبایل تمام‌عرض می‌شوند.

- KPI‌ها جمع‌وجوی تر.

- کارت کمتر می‌شود و متن توضیح کوچک‌تر.

## خلاصه عملکرد

این CSS یک تم تیره (Dark Theme) راستبهچپ برای داشبورد IoT می‌سازد:

- پس زمینه گرادیانی و کارت‌ها نیمه‌شفاف با افکت شیشه‌ای.
- جدول‌ها، KPI‌ها و دکمه‌ها استایل ویژه دارند.
- دکمه‌های ری‌استارت/خاموش ظاهر مشکی خاص با حالت hover/active.
- متن‌ها و عنوان‌ها سفید و خوانا.
- روی موبایل همه اجزا واکنش‌گرا و جمع‌وجورتر نمایش داده می‌شوند.

//assets/app.js

```
const IOT} <= () =
```

یک IIFE (تابع فوراً‌اجرا) که یک شیء سراسری به نام IOT برمی‌گرداند و بقیه توابع را درون خودش کپسوله می‌کند.

```
const fmt = d => new Date(d).toLocaleString('fa-IR');
```

کمکی برای فرمات تاریخ/ساعت به محلی «fa-IR»

## Dashboard KPIs

```
async function pollHealth} ()
try}

const r = await fetch('api/system_health.php', { cache: 'no-store' });
if (!r.ok) return;
```

```
const j = await r.json();()
```

هر بار سلامت سیستم را از API می‌گیرد (بدون کش مرورگر).

```
function colorize(id, val){
 const el = document.getElementById(id);
 if (!el) return;
 el.className = 'kpi-value';
 let cls = 'good';
 if (val >= 80) cls = 'bad';
 else if (val >= 50) cls = 'warn';
 el.classList.add(cls);
 el.textContent = (id === 'temp')
 ? {Number(val).toFixed(1)}°C
 : Number(val).toFixed(1)
}
```

تابع داخلی: مقدار را رنگبندی می‌کند (سبز/زرد/قرمز) و متن مناسب (%) یا (°C) می‌نویسد.

```
colorize('cpu', j.cpu_percent ?? 0);
colorize('ram', j.ram_percent ?? 0);
colorize('temp', j.cpu_temp_c ?? 0);
```

سه KPI اصلی: RAM, CPU, دما.

```
const netEl = document.getElementById('net');
if (netEl){
```

```

const ip = j.ip_addr; '—' ??

const rssi = (j.wifi_rssi ?? '—') + (j.wifi_rssi !== null && j.wifi_rssi !== undefined ? ' dBm' :

");

netEl.replaceChildren(document.createTextNode(` ${ip} / RSSI ${rssi}`));

{

```

نمایش IP و RSSI در یک باکس شبکه.

```
{
 catch (e) { /* ignore { /*
```

خطاهای شبکه نادیده گرفته می‌شوند (UI را خراب نکند).

(فهرست رویدادها با فیلتر/صفحه‌بندی) Events

```
let EV_STATE = { page: 1, per: 50 };

let EV_TIMER = null;

let EV_BUSY = false;
```

وضعیت جاری صفحه رویدادها: صفحه، اندازه صفحه، تایمر رفرش خودکار؛ و پرچم «مشغول بودن» تا درخواست همپوشان نشود.

```
function applyEventFilters} ()

EV_STATE.page = 1;

refreshEvents(false);

{
```

وقتی کاربر فیلتر عوض کرد: از صفحه‌ی ۱ مجدد بارگذاری.

```
async function refreshEvents(first){

 if (EV_BUSY) return;
```

```
EV_BUSY = true;
```

از هم زمانی جلوگیری؛ فقط یک درخواست فعال.

```
const btn = document.querySelector('button.btn.btn-sm.btn-primary');

const evPerfEl = document.getElementById('evPerf');

const t0 = performance.now();
```

دکمه‌ی «اعمال فیلتر» و المان نمایش عملکرد؛ شروع تایمر سمت کلاینت.

```
try{

if (btn){

 btn.classList.add('btn-filter-fixed');

 const w = btn.getBoundingClientRect().width;

 btn.style.width = Math.ceil(w) + 'px';

 btn.disabled = true;

 btn.dataset._orig = btn.innerHTML;

 btn.innerHTML = '</span <';

 اعمال فیلتر';

}

if (evPerfEl) evPerfEl.textContent = 'در حال جستجو...' ;
```

قفل کردن عرض دکمه و نشان دادن spinner تا پرش UI رخ ندهد.

```
const params = new URLSearchParams();

const s = document.getElementById('start')?.value;" ??

const e = document.getElementById('end')?.value;" ??

const q = (document.getElementById('search')?.value ?? "").trim();
```

```
const confMin = document.getElementById('confMin')?.value;" ??
const confMax = document.getElementById('confMax')?.value;" ??
 .confidence
خواندن فیلترها از فرم: شروع/پایان، جستجو، بازه‌ی
```

```
const perSel = document.getElementById('evPer');
if (perSel){
 const perVal = parseInt(perSel.value || '50', 10);
 if (!isNaN(perVal)) EV_STATE.per = perVal;
}
```

اندازه‌ی صفحه (per page) از انتخاب کاربر.

```
if (q) params.set('q', q);
if (s) params.set('start', s);
if (e) params.set('end', e);
if (confMin !== "") params.set('conf_min', confMin);
if (confMax !== "") params.set('conf_max', confMax);
params.set('page', EV_STATE.page);
params.set('per', EV_STATE.per);
```

ساختQueryString برای API events\_list.php

```
const r = await fetch('api/events_list.php?' + params.toString(), { cache: 'no-store' });
const txt = await r.text();
let j; try { j = JSON.parse(txt); } catch { j = null; }
if (!r.ok || !j || j.ok === false)
 console.error('events_list error:', txt);
```

```

alert + ('خطا در واکشی رویدادها. لطفاً صفحه را رفرش کنید.');

return;

{

```

واکشی JSON، خطاهای هشدار به کاربر و لگ در کنسول.

```

const tb = document.getElementById('eventsFull'); if (!tb) return;

tb.innerHTML =

(j.rows || []).forEach((x, i) => {

 const tr = document.createElement('tr');

 tr.innerHTML=>

 > td>${(EV_STATE.page - 1) * EV_STATE.per + i + 1}</td<
 > td>${fmt(x.ts)}</td<
 > td>${x.label}</td<
 > td>${Number(x.confidence).toFixed(1)}%</td<
 > td>${x.node_id}</td;<

 tb.appendChild(tr);

};)

```

رندر جدول نتایج (شماره‌ی ردیف با احتساب صفحه).

```

EV_STATE.page = j.page ?? EV_STATE.page;
EV_STATE.per = j.per ?? EV_STATE.per;
const pages = j.pages ?? 1;
const total = j.total ?? 0;

```

همگام‌سازی وضعیت با پاسخ سرور.

```

const elPage = document.getElementById('evPage');

const elPages = document.getElementById('evPages');

const elTotal = document.getElementById('evTotal');

if (elPage) elPage.textContent = EV_STATE.page;

if (elPages) elPages.textContent = pages;

if (elTotal) elTotal.textContent = total;

```

به روزرسانی pagination (صفحه‌ی فعال/کل صفحات/تعداد نتایج).

```

const prevBtn = document.getElementById('evPrev');

const nextBtn = document.getElementById('evNext');

if (prevBtn){

 prevBtn.disabled = (EV_STATE.page <= 1);

 prevBtn.onclick = () => { if (EV_STATE.page > 1) { EV_STATE.page--; refreshEvents(false); }};

}

if (nextBtn){

 nextBtn.disabled = (EV_STATE.page >= pages);

 nextBtn.onclick = () => { if (EV_STATE.page < pages) { EV_STATE.page++; refreshEvents(false); }};

}

```

ناوبری صفحات: دکمه‌ی قبلی/بعدی با فعال/غیرفعال سازی.

```

if (perSel && !perSel._bound)

 perSel.addEventListener('change', () => { EV_STATE.page = 1; refreshEvents(false); });

```

```
perSel._bound = true;
{
```

تغییر اندازه‌ی صفحه → بازنشانی به صفحه‌ی ۱ (یکبار bind شود).

```
const startEl = document.getElementById('start');
const endEl = document.getElementById('end');
const qEl = document.getElementById('search');
const cminEl = document.getElementById('confMin');
const cmaxEl = document.getElementById('confMax');
[startEl, endEl, qEl, cminEl, cmaxEl].forEach(el) <=
if (el && !el._bound){
 el.addEventListener('change', () => { EV_STATE.page = 1; refreshEvents(false); });
 if (el.tagName === 'INPUT'){
 el.addEventListener('keyup', (ev) => { if (ev.key === 'Enter') { EV_STATE.page = 1; refreshEvents(false); } });
 }
 el._bound = true;
}
```

هر تغییر فیلتر (یا Enter) → بارگذاری مجدد از صفحه‌ی ۱؛ یکبار bind.

```
const t1 = performance.now();
const clientMs = Math.max(0, t1 - t0);
const hasDb = (typeof j.bench_ms === 'number');
const dbStr = hasDb ? `${j.bench_ms.toFixed(0)} ms` : `
```

```

const totalStr= `${clientMs.toFixed(0)} ms;`

let indexStr;" =

if (j.explain && j.explain[0]) {

 const k = j.explain[0].key || j.explain[0].partitions || null;

 if (k) indexStr = ` · index=${k};`

}

if (evPerfEl) {

 evPerfEl.textContent = `عملکرد کوئری: کل = ${dbStr} ${totalStr}${indexStr};`

}

```

نمایش عملکرد:

زمان اجرای SELECT در سرور (اگر EXPLAIN ANALYZE/EXPLAIN API با bench\_ms زمان اجرای

زمان کل سمت کلاینت (شبکه + رندر).

نکته‌ی مهم ایندکس: اگر API خروجی explain داده باشد، با j.explain[0].key نام ایندکسی که واقعاً استفاده شده را نشان می‌دهد (index...). اینجا دقیقاً برای تأکید روی استفاده شدن ایندکس کوئری اضافه شده.

```

if (first) {

 if (EV_TIMER) clearInterval(EV_TIMER);

 EV_TIMER = setInterval(() => refreshEvents(false), 5000);

}

```

اگر اولین بار است، رفرش خودکار هر ۵ ثانیه.

```

 {
 catch (e) (
 console.error(e);
 alert('خطای غیرمنتظره در بارگذاری رویدادها')
 }
 finally{
 EV_BUSY = false;
 if (btn)
 btn.disabled = false;
 btn.innerHTML = btn.dataset._orig ?? '<i class="bi bi-funnel"></i>'; < اعمال فیلتر'
 setTimeout(() => { btn.style.width = ''; }, 50);
 }
 }
}

```

آزادسازی وضعیت و برگرداندن دکمه به حالت عادی.

### (Kill + Processes فهرست پردازهای فهرست

```

async function refreshProcesses() {
 try{
 const r = await fetch('api/processes_list.php', { cache: 'no-store' });
 if (!r.ok) return;
 const j = await r.json();
 }
}

```

دریافت فهرست پردازهای API از

```

const tb = document.getElementById('procFull'); if (!tb) return;
tb.innerHTML = =

```

جدول را خالی و ردیفهای تازه می‌سازد.

```
(j.rows || []).forEach((x, i) => {
 const tr = document.createElement('tr');
 const btn = `
 > button class="btn btn-sm btn-danger" onclick="IOT.killPid(${x.pid}, '${x.user}')<"
 > i class="bi bi-x-circle"></i<
 /> button;`<

 tr.innerHTML =
 > td>${i + 1}</td<
 > td>${x.pid}</td<
 > td>${x.user ?? '—'}</td<
 > td>${x.name}</td<
 > td>${x.cpu_percent}</td<
 > td>${x.mem_percent}</td<
 > td>${fmt(x.created_at)}</td<
 > td>${btn}</td;`<
 tb.appendChild(tr);
};{
```

```
{ catch (e) { /* ignore */ }
```

پرکردن جدول پردازه‌ها.

```

async function killPid(pid, user) {
 if(!confirm(`آیا از خاتمه پردازه ${pid} (${user}) مطمئن هستید؟`)) return;
 try {
 const form = new FormData();
 form.append('pid', pid);
 }

```

تایید کاربر و آماده سازی POST.

```

const r = await fetch('api/process_kill.php', { method: 'POST', body: form, cache: 'no-store' });

```

```
const txt = await r.text();
```

```
let j; try { j = JSON.parse(txt); } catch { j = null; }
```

پاسخ را متن می‌گیرد و سپس JSON می‌کند (برای هندل پاسخ‌های غیر JSON هم).

```

if (!r.ok || !j || !j.ok)

const msg = j?.message || j?.error || txt; ||
'نامشخص' ||

alert(`خطا در Kill:\n${msg}`);

if (j?.stderr) console.error('stderr:', j.stderr);

return;

{
 alert(`پردازه ${pid} خاتمه یافت.`)
 refreshProcesses();
}

 catch (e) {
 alert(`پردازه ${pid} خاتمه یافت.`)
 }
}

```

```
{
{
 پیام موفق/ناموفق و رفرش لیست.
}
```

```
Metrics (نمايش آخرين ۲۰ متریک)
async function loadMetrics() {
 try {
 const r = await fetch('api/cpu_metrics_list.php', { cache: 'no-store' });
 if (!r.ok) return;
 const j = await r.json();
 }
 گرفتن فهرست متریکها.
}
```

```
const tb = document.getElementById('metricsT'); if (!tb) return;
const st = document.getElementById('st'),
 ip = document.getElementById('ip'),
 rssI = document.getElementById('rssI'),

tb.innerHTML =
(j.rows || []).forEach(x) =<
 const tr = document.createElement('tr');
 tr.innerHTML =
 '> td>${fmt(x.created_at)}</td<
 > td>${x.cpu_percent}</td<
 > td>${x.ram_percent}</td<
```

```

> td>${x.cpu_temp_c ?? '—'}</td<
> td>${x.ip_addr ?? '—'}</td<
> td>${x.wifi_rssi ?? '—'}</td;><
tb.appendChild(tr);
;({

```

رندر جدول متريکها.

```

if (j.rows?.length){

 const last = j.rows[0]

 if (st) st.textContent='نلاين' =
 if (ip) ip.textContent = last.ip_addr;'—' ??
 if (rss) rss.textContent = (last.wifi_rssi ?? '—') + (last.wifi_rssi ? ' dBm' : "");

 {
 else}

 if (st) st.textContent='—' =
 if (ip) ip.textContent;'—' =
 if (rss) rss.textContent;'—' =

 {
 catch (e){ /* ignore{ /*


```

خلاصهی وضعیت کنار جدول بر اساس آخرين ردیف.

مشخصات سیستم (System Specs)

```

async function loadSpecs() {
 try{

```

```
const r = await fetch('api/system_specs.php', { cache: 'no-store' });

if (!r.ok) return;

const j = await r.json();
```

واکشی مشخصات سیستم.

```
const set = (id, val) => { const el = document.getElementById(id); if (el) el.textContent =
(val ?? '—'); };
```

```
set('dev', j.device_model);

set('host', j.hostname);

set('os', j.os);

set('kernel', j.kernel);

set('arch', j.arch);

set('cpu', j.cpu_model);

set('cores', j.cpu_cores);
```

جاگذاری مقادیر پایه.

```
set('ram_total', j.mem_total_mb != null ? `${j.mem_total_mb} MB` : '—');

set('ram_used', j.mem_used_mb != null ? `${j.mem_used_mb} MB` : '—');

set('ram_free', j.mem_free_mb != null ? `${j.mem_free_mb} MB` : '—');

const ramBar = document.getElementById('ram_bar');

if (ramBar && j.mem_total_mb && j.mem_used_mb != null)

 const p = Math.max(0, Math.min(100, (j.mem_used_mb * 100) / j.mem_total_mb));

 ramBar.style.width = p.toFixed('.' + (0))

 ramBar.textContent = p.toFixed('.' + (0))
```

```

ramBar.classList.remove('bg-success', 'bg-warning', 'bg-danger');

if (p >= 80) ramBar.classList.add('bg-danger');

else if (p >= 50) ramBar.classList.add('bg-warning');

else ramBar.classList.add('bg-success');

{

```

نمایش نوار پیشرفت RAM و رنگبندی آن.

```

set('disk_total', j.disk_total_mb != null ? `${j.disk_total_mb} MB` : '—');

set('disk_used', j.disk_used_mb != null ? `${j.disk_used_mb} MB` : '—');

set('disk_free', j.disk_free_mb != null ? `${j.disk_free_mb} MB` : '—');

const diskBar = document.getElementById('disk_bar');

if (diskBar && j.disk_total_mb && j.disk_used_mb != null)

 const p = Math.max(0, Math.min(100, (j.disk_used_mb * 100) / j.disk_total_mb));

 diskBar.style.width = p.toFixed('%.0' + (0))

 diskBar.textContent = p.toFixed('%.0' + (0))

{
}

{
 catch (e) { /* ignore */ }
}
```

نمایش نوار پیشرفت دیسک.

KPI کوچک در صفحه‌ی سیستم

```

async function pollHealthKPI(prefix = "")

try{

const r = await fetch('api/system_health.php', { cache: 'no-store' });


```

```
if (!r.ok) return;
const j = await r.json();
```

گرفتن سلامت سیستم (مثل همان Dashboard) اما با شناسه‌های prefixed (برای چند بакс).

```
function colorizeEl(el, val, isTemp = false){

if (!el) return;

el.className = 'kpi-value';

let cls = 'good';

if (!isTemp){

if (val >= 80) cls = 'bad';

else if (val >= 50) cls = 'warn';

el.classList.add(cls);

el.textContent = isNaN(val) ? '—' : `${Number(val).toFixed(1)}%`;

}
else{

if (val >= 75) cls = 'bad';

else if (val >= 60) cls = 'warn';

el.classList.add(cls);

el.textContent = isNaN(val) ? '—' : `${Number(val).toFixed(1)}°C`;

}
```

همان منطق رنگبندی با آستانه‌های متفاوت برای دما.

```
colorizeEl(document.getElementById(prefix + 'cpuK'), j.cpu_percent ?? NaN, false);
colorizeEl(document.getElementById(prefix + 'ramK'), j.ram_percent ?? NaN, false);
```

```
colorizeEl(document.getElementById(prefix + 'tmpK'), j.cpu_temp_c ?? NaN, true);
```

```
const ipEl = document.getElementById(prefix + 'ipK');

if (ipEl) ipEl.textContent = 'IP: ' + (j.ip_addr ?? '—');

const rs = document.getElementById(prefix + 'rssK');

if (rs) rs.textContent = 'RSSI: ' + ((j.wifi_rssi ?? '—') + (j.wifi_rssi ? ' dBm' : ""));

{ catch (e) { /* ignore */
```

```
{
```

مقداردهی عناصر KPI کوچک.

(ریاستارت/خاموش) Power Controls

```
async function sysPower(action)

if (!['reboot', 'shutdown'].includes(action)) return;
```

const fa = action === 'reboot' ? 'ریاستارت' : 'خاموش'

if(!) confirm { سیستم مطمئن هستید؟(`) (آیا از \$

تایید کاربر و اعتبارسنجی نوع عمل.

```
const btn = typeof event !== 'undefined' ? event.currentTarget : null;

try}

if (btn) } btn.disabled = true; btn.dataset._orig = btn.innerHTML; btn.innerHTML =
 ;'...< در حال ارسال
```

```
const fd = new FormData();
```

```

fd.append('action', action);

const r = await fetch('api/system_power.php', { method: 'POST', body: fd, cache: 'no-store' });

const txt = await r.text();

let j; try { j = JSON.parse(txt); } catch { j = null; }

if (!r.ok || !j?.ok)

 alert) + ' | | ' + 'نامشخص' + 'خطا: ' + j?.error || txt;

{
 else}

 alert} $ ارسال شد.' + '(`` دستور fa;

{

{
 catch (e) (

 alert; ' + 'خطای شبکه/سرور';

{
 finally}

if (btn) { btn.disabled = false; btn.innerHTML = btn.dataset._orig || btn.innerHTML; }

{
{
}

```

ارسال درخواست به API و مدیریت UI دکمه.

### Init ها و خروجی مازول

```

function initDashboard() { pollHealth(); setInterval(pollHealth, 3000); }

function initEventsPage() { refreshEvents(true); }

function initProcessesPage() { refreshProcesses(); setInterval(refreshProcesses, 5000); }

function initMetricsPage() { loadMetrics(); setInterval(loadMetrics, 5000); }

function initSystemPage() { loadSpecs(); pollHealthKPI(""); setInterval(() =>
pollHealthKPI(""), 3000); }

```

توابع راهانداز هر صفحه (یکبار بارگذاری و سپس بازهای رفرش).

```
return}

initDashboard,

initEventsPage,

initProcessesPage,

initMetricsPage,

initSystemPage,

refreshEvents,

applyEventFilters,

killPid,

sysPower
;{
;(){
```

در دسترس گذاشتن API عمومی شیء IoT.

خلاصه‌ی عملکرد

این فایل همه‌ی منطق فرانت‌اند را برای داشبورد، رویدادها (با فیلتر/صفحه‌بندی)، پردازه‌ها (به همراه Kill)، متريک‌ها، مشخصات سیستم و کنترل‌های پاور مدیریت می‌کند.

برای رویدادها:

با events\_list.php صحبت می‌کند، جدول را می‌سازد، صفحه‌بندی/فیلترها را اعمال می‌کند، و عملکرد کوئری را نشان می‌دهد.

اگر API خروجی explain/bench بدهد، این کد نام ایندکس استفاده شده را با index=j.explain[0].key ... از استخراج و کنار زمان ها نمایش می دهد—این دقیقاً جایی است که روی ایندکس کوئری تاکید می شود.

سایر بخش ها داده ها را از API های مربوطه می گیرند، DOM را به روز می کنند و بازه ای رفرش می شوند.

### docker-compose.yml:

services:

iot-audio:

build:

context: ./audio

container\_name: iot-audio

restart: unless-stopped

environment:

# DB

DB\_HOST: 127.0.0.1 # اگر MySQL را بگذار/IP ابیرون از کانتینر است، نام سرویس واقعی را بگذار

DB\_NAME: iot

DB\_USER: iotuser

DB\_PASS: iotpass

# MQTT (+ TLS)

MQTT\_HOST: 192.168.100.102

MQTT\_PORT: 8883

MQTT\_USER: Ali

MQTT\_PASS: EmbeddedProject

```
MQTT_BASE_TOPIC: iot/pi-99
MQTT_TLS_CA: /run/certs/ca.pem # الزامی برای TLS
```

# داری (کلاینت هم گواهی/کلید دارد) این دو خط را از کامنت دربیار اگر mTLS:

```
MQTT_TLS_CERT: /run/certs/client.crt
MQTT_TLS_KEY: /run/certs/client.key
```

# ضبط

USB\_INDEX: "2"

DURATION: "10"

devices:

```
- "/dev/snd:/dev/snd" # دسترسی به میکروفون
```

volumes:

```
- ./audio/app:/app # کد سرویس iot-audio
- ./metrics/certs:/run/certs:ro # گواهی‌ها فقط خواندنی
```

group\_add:

```
- "audio" # برای ALSA
```

iot-metrics:

build:

context: ./metrics

container\_name: iot-metrics

restart: unless-stopped

volumes:

```
- ./metrics/app:/app
- ./metrics/certs:/run/certs:ro # گواهی‌ها فقط خواندنی
```

environment:

```
MQTT_HOST: 192.168.100.102
MQTT_PORT: 8883
MQTT_USER: Ali
MQTT_PASS: EmbeddedProject
MQTT_BASE_TOPIC: iot/pi-99
INTERVAL_SEC: "3"
```

# TLS

```
MQTT_TLS_CA: /run/certs/ca.pem # الزامی برای TLS
اگر mTLS داری:
MQTT_TLS_CERT: /run/certs/client.crt
MQTT_TLS_KEY: /run/certs/client.key
```

توضیح :

:services

بخش سرویس‌ها در docker compose (هر سرویس = یک کانتینر/اپ مستقل).

iot-audio:

نام سرویس اول (کانتینر پردازش اصیل صوت).

build:

context: ./audio

ایمیج از دایرکتوری `audio/` ساخته می‌شود (`Dockerfile` همان‌جاست).

container\_name: iot-audio

نام ثابت کانتینر در داکر: `.iot-audio`

restart: unless-stopped

سیاست ری‌استارت: خودکار بالا می‌آید مگر اینکه دستی استاپ‌ش کرده باشد.

environment:

متغیرهای محیطی داخل کانتینر.

# DB

DB\_HOST: 127.0.0.1

DB\_NAME: iot

DB\_USER: iotuser

DB\_PASS: iotpass

تنظیمات دیتابیس (مقداردهی از محیط).

توجهِ توصیفی: ۱۲۷.۰.۰.۱ داخل کانتینر یعنی لوپ‌بک همان کانتینر.

```
MQTT
```

```
MQTT_HOST: 192.168.100.102
```

```
MQTT_PORT: 8883
```

```
MQTT_USER: Ali
```

```
MQTT_PASS: EmbeddedProject
```

```
MQTT_BASE_TOPIC: iot/pi-99
```

تنظیمات اتصال MQTT (هاست/پورت/کاربر/پسورد/بیس تاپیک).

```
ضبط
```

```
USB_INDEX: "2"
```

```
DURATION: "10"
```

پارامترهای اپ صوت: ایندکس کارت/میکروفون USB و طول هر ضبط (ثانیه) به صورت رشته.

```
devices:
```

```
/" - dev/snd:/dev/snd # دسترسی به میکروفون "
```

مَپ کردن دیوایس صدای میزبان به داخل کانتینر (برای ALSA/ضبط).

```
volumes:
```

```
/. - audio/app:/app # کد + مدل (اگر مدل سنگینه، می‌توانی جدا مونت کنی)
```

مَپ دایرکتوری کد/مدل میزبان به مسیر app داخل کانتینر (برای دو/آپدیت سریع).

```
نیاز به دسترسی بیشتری داشت: # اگر ALSA
```

```
group_add:
```

" - audio"

افزودن کانتینر به گروه سیستم audio (برای مجوز دسترسی به کارت صدا).

iot-metrics:

نام سرویس دوم (جمع‌آوری متريک‌هاي سیستم و ارسال روی MQTT).

build:

context: ./metrics

ایمیج از دایرکتوری metrics/. ساخته می‌شود.

container\_name: iot-metrics

نام ثابت کانتینر: iot-metrics

restart: unless-stopped

همان سیاست ری‌استارت خودکار.

environment:

MQTT\_HOST: 192.168.100.102

MQTT\_PORT: 8883

MQTT\_USER: Ali

MQTT\_PASS: EmbeddedProject

MQTT\_BASE\_TOPIC: iot/pi-99

INTERVAL\_SEC: "3"

متغیرهای محیطی: تنظیمات اتصال MQTT و بازه‌ی نمونه‌برداری متريک‌ها (هر ۳ ثانیه).

volumes:

/.- metrics/app:/app

مَبِ کِدِ سرویسِ متريک‌ها به /app داخل کانتينر.

خلاصه‌ی عملکرد

این Compose دو سرویس راه می‌اندازد:

iot-audio: ايميج از /audio ساخته می‌شود، به کارت صدای میزبان (/dev/snd) دسترسی دارد، با محیط DB/MQTT و پارامترهای ضبط کار می‌کند، و با گروه audio اجرا می‌شود.

iot-metrics: ايميج از /metrics ساخته می‌شود، هر ۳ ثانیه متريک‌ها را جمع کرده و به MQTT (هاست/پورت مشخص) می‌فرستد. هر دو سرویس با restart: unless-stopped پايدار نگه داشته می‌شوند و کدها از میزبان به /app مَبِ شده‌اند.

مانیتور منابع (resource\_monitor\_mqtt.py)

```
sudo tee /etc/systemd/system/iot-resource-monitor.service >/dev/null <<'UNIT'
```

[Unit]

Description=IoT Resource Monitor (MQTT)

After=network-online.target

Wants=network-online.target

[Service]

Type=simple

WorkingDirectory=/opt/iot/python

```
ExecStart=/usr/bin/python3 /opt/iot/python/resource_monitor_mqtt.py
Restart=always
RestartSec=3
Environment=PYTHONUNBUFFERED=1
محدودیت‌های سبک اختیاری
Nice=5
CPUQuota=80%
سخت‌سازی پیشنهادی
NoNewPrivileges=true
ProtectSystem=full
ReadWritePaths=/opt/iot
ProtectHome=true
PrivateTmp=true
```

#### [Install]

```
WantedBy=multi-user.target
```

#### UNIT

```
:system كد (detect_audio_yamnet_db_ML.py) تشخیص صدا
```

```
sudo tee /etc/systemd/system/iot-detect-audio.service >/dev/null <<'UNIT'
```

#### [Unit]

```
Description=Audio Event Detection (YAMNet)
```

```
After=network-online.target sound.target mysql.service mariadb.service
```

```
Wants=network-online.target sound.target
```

#### [Service]

```
Type=simple
WorkingDirectory=/opt/iot/python
ExecStart=/usr/bin/python3 /opt/iot/python/detect_audio_yamnet_db_ML.py
Restart=always
RestartSec=5
Environment=PYTHONUNBUFFERED=1
After=mosquitto.service
Wants=mosquitto.service
```

# ساخت سازی پیشنهادی

```
NoNewPrivileges=true
ProtectSystem=full
ReadWritePaths=/opt/iot
ProtectHome=true
PrivateTmp=true
StartLimitIntervalSec=0
```

[Install]

```
WantedBy=multi-user.target
UNIT
```

: توضیح

۱. سرویس مانیتور منابع (**iot-resource-monitor.service**)

Description=IoT Resource Monitor (MQTT)

After=network-online.target

Wants=network-online.target

بخش **Unit** توضیح سرویس و وابستگی‌ها.

فقط بعد از بالا آمدن کامل شبکه اجرا شود.

اعلام می‌کند که به شبکه نیاز دارد.

Type=simple

WorkingDirectory=/opt/iot/python

ExecStart=/usr/bin/python3 /opt/iot/python/resource\_monitor\_mqtt.py

Restart=always

RestartSec=3

Environment=PYTHONUNBUFFERED=1

فرآیند مستقیم همان دستور است.

/opt/iot/python. مسیر کاری **WorkingDirectory**

اجرای اسکریپت resource\_monitor\_mqtt.py با پایتون<sup>۳</sup>.

اگر کوش کند دوباره اجرا شود.

۳ ثانیه فاصله برای ریاستارت.

خروجی‌ها بدون بافر باشند (برای لاغ لحظه‌ای).

# محدودیت‌های سبک اختیاری

# Nice=5

# CPUQuota=80%

تنظیمات اختیاری برای محدود کردن اولویت CPU یا سهمیه.

# سخت‌سازی پیشنهادی

```
NoNewPrivileges=true
ProtectSystem=full
ReadWritePaths=/opt/iot
ProtectHome=true
PrivateTmp=true
```

#### • systemd امنیت

- **NoNewPrivileges=true:** پردازه نمی‌تواند سطح دسترسی بیشتری بگیرد.
- **ProtectSystem=full:** بیشتر سیستم فقط خواندنی می‌شود.
- **ReadWritePaths=/opt/iot:** فقط این مسیر اجازه نوشتن دارد.
- **ProtectHome=true:** دسترسی به Home کاربران بسته می‌شود.
- **PrivateTmp=true:** دایرکتوری /tmp/ اختصاصی.

```
WantedBy=multi-user.target
```

#### • سرویس در حالت چندکاربره (runlevel) پیش‌فرض لینوکس سرور (فعال می‌شود).

## ۲. سرویس تشخیص صدا (iot-detect-audio.service)

```
Description=Audio Event Detection (YAMNet)
```

```
After=network-online.target sound.target mysql.service mariadb.service
```

```
Wants=network-online.target sound.target
```

• توضیح: تشخیص رویداد صوتی با YAMNet.

• اجرا بعد از بالا آمدن شبکه، صدا، و دیتابیس (MySQL/MariaDB).

• Wants: شبکه و سرویس صوتی را درخواست می‌کند.

```
Type=simple
```

```
WorkingDirectory=/opt/iot/python
```

```
ExecStart=/usr/bin/python3 /opt/iot/python/detect_audio_yamnet_db_ML.py
Restart=always
RestartSec=5
Environment=PYTHONUNBUFFERED=1
After=mosquitto.service
Wants=mosquitto.service
```

- مشابه قبلی: اسکریپت صوتی اجرا می‌شود.
- فاصله ری‌استارت ۵ ثانیه است.
- امکان اضافه کردن وابستگی به Mosquitto در صورت استفاده از MQTT با خطهای کامنت شده.

# سخت‌سازی پیشنهادی

```
NoNewPrivileges=true
ProtectSystem=full
ReadWritePaths=/opt/iot
ProtectHome=true
PrivateTmp=true
StartLimitIntervalSec=0
```

WantedBy=multi-user.target

- همان سیاست‌های امنیتی.
- اگر uncomment شود، محدودیت تعداد ری‌استارت‌ها را برمی‌دارد.

• سرویس هنگام بوت فعال می‌شود.

خلاصه عملکرد

- **iot-resource-monitor.service** اسکریپت مانیتور منابع CPU, RAM, دما، شبکه‌را به صورت دائمی اجرا می‌کند و داده‌ها را روی MQTT می‌فرستد.

• **iot-detect-audio.service** سکریپت تشخیص صدا با YAMNet را دائمی می‌کند و وابسته به شبکه، صدا و دیتابیس است.

• هر دو سرویس با **systemd** به طور خودکار پس از بوت اجرا می‌شوند، در صورت خطا ریاستارت می‌شوند، و تنظیمات امنیتی پایه برای محدود کردن دسترسی آن‌ها اعمال شده است.

## Audio

### Dockerfile:

```
برای ARM64
```

```
FROM arm64v8/python:3.10-slim
```

```
ENV DEBIAN_FRONTEND=noninteractive \
```

```
PYTHONDONTWRITEBYTECODE=1 \
```

```
PYTHONUNBUFFERED=1
```

```
وابستگی‌های سیستمی صدا و علمی
```

```
RUN apt-get update && apt-get install -y --no-install-recommends \
```

```
libportaudio2 libsndfile1 alsa-utils \
```

```
libatlas-base-dev gfortran build-essential \
```

```
&& rm -rf /var/lib/apt/lists/*
```

```
کار دایرکتوری
```

```
WORKDIR /app
```

```
نیازمندی‌های پایتون
```

```
COPY requirements.txt /app/requirements.txt
```

```
RUN pip install --no-cache-dir -r requirements.txt
```

```
کد و مدل‌ها
```

```
COPY app/ /app/
```

```
داخل کانتینر ALSA برای
```

```
(از میزبان گرفت volume در صورت نیاز: تنظیمات را می‌توان با) #
```

```
ENV PYTHONPATH=/app \
```

```
TZ=Asia/Tehran
```

```
Healthcheck ساده: وجود فایل مدل و اجرای کوتاه python
```

```
HEALTHCHECK --interval=30s --timeout=10s --retries=5 \
```

```
CMD [-f /app/yamnet.h5] || exit 1
```

```
اجرای برنامه
```

```
CMD ["python3", "/app/detect_audio_yamnet_db.py"]
```

توضیح :

```
برای ARM64
```

```
FROM arm64v8/python:3.10-slim
```

پایه‌ی ایمیج: پایتون 3.10 لایت برای معماری ARM64 (مثلًا Orange Pi, Raspberry Pi 64bit)

همه‌ی لایه‌ها روی این بیس ساخته می‌شوند.

```
ENV DEBIAN_FRONTEND=noninteractive\
```

```
PYTHONDONTWRITEBYTECODE=1\
```

```
PYTHONUNBUFFERED=1
```

بدون پرسش تعاملی apt: نصب

PYTHONDONTWRITEBYTECODE=1 .pyc: فایل‌های کوچک‌تر و تمیزتر.

PYTHONUNBUFFERED=1 : خروجی پایتون بدون بافر (لاگ‌ها فوری).

# وابستگی‌های سیستمی صدا و علمی

```
RUN apt-get update && apt-get install -y --no-install-recommends\
```

```
libportaudio2 libsndfile1 alsa-utils\
```

```
libatlas-base-dev gfortran build-essential\
```

```
&& rm -rf /var/lib/apt/lists*/
```

نصب پکیج‌های سیستمی مورد نیاز:

برای ورودی/خروجی صدا PortAudio, ALSA.( libportaudio2, libsndfile1, alsa-utils )

برای کتابخانه‌های علمی NumPy/Scipy ( libatlas-base-dev, gfortran, build-essential ) و کامپایل.

پاک‌کردن لیست‌های apt برای کوچک شدن ایمیج.

# کار دایرکتوری

```
WORKDIR /app
```

مسیر کاری پیشفرض داخل کانتینر را /app تنظیم می‌کند.

```
نیازمندی‌های پایتون
```

```
COPY requirements.txt /app/requirements.txt
```

```
RUN pip install --no-cache-dir -r requirements.txt
```

فایل نیازمندی‌ها را کپی می‌کند.

همه‌ی پکیج‌های پایتونی پروژه را نصب می‌کند (بدون کش → ایمیج سبک‌تر).

```
کد و مدل‌ها
```

```
COPY app/ /app/
```

کل سورس پروژه (و مدل‌ها مثل `yamnet.h5`) در `app/` باشند) را به داخل ایمیج می‌آورد.

```
داخل کانتینر #ALSA برای
```

از میزبان گرفت) `#volume` (در صورت نیاز: تنظیمات را می‌توان با

```
ENV PYTHONPATH=/app\
```

```
TZ=Asia/Tehran
```

`PYTHONPATH=/app`: تا ماثول‌های داخل `app/` به راحتی ایمپورت شوند.

`TZ=Asia/Tehran`: منطقه‌ی زمانی داخل کانتینر.

```
#Healthcheck ساده: وجود فایل مدل و اجرای کوتاه python
```

```
HEALTHCHECK --interval=30s --timeout=10s --retries=5\
```

```
CMD [-f /app/yamnet.h5] || exit 1
```

سلامت کانتینر هر ۳۰ ثانیه بررسی می شود.

اگر فایل مدل yamnet.h5 وجود نداشته باشد → وضعیت .unhealthy

# اجرای برنامه

```
CMD ["python3", "/app/detect_audio_yamnet_db.py"]
```

فرمان پیش فرض کانتینر: اجرای اسکریپت اصلی تشخیص صوت.

خلاصه‌ی عملکرد

این Dockerfile یک ایمیج ARM64 سبک می‌سازد، و استگی‌های صوتی/علمی و پکیج‌های پایتون پروژه را نصب می‌کند، کد و مدل را به app کپی می‌کند، متغیرهای محیطی لازم را تنظیم می‌کند، با Healthcheck وجود مدل را می‌سنجد، و در نهایت اسکریپت تشخیص صدا را با پایتون اجرا می‌کند.

Restart

burn\_ram.py:

```
usr/bin/env python3/#!#
```

```
import argparse, time, os, sys
```

```
PAGE = 4096
```

```
MB = 1024 * 1024
```

```
:()def meminfo_mb
```

```
"""(MemTotal_MB, MemAvailable_MB)""""برگرداندن
```

```
total_kb = avail_kb = 0
```

```

:with open("/proc/meminfo") as f
:for line in f
:if line.startswith("MemTotal:")
total_kb = int(line.split()[1])
:elif line.startswith("MemAvailable:")
avail_kb = int(line.split()[1])
:if not avail_kb: # fallback
:with open("/proc/meminfo") as f
:for line in f
:if line.startswith("MemFree:")
avail_kb = int(line.split()[1])
break
return total_kb // 1024, avail_kb // 1024

():def rss_mb
:try
:with open("/proc/self/status") as f
:for line in f
:if line.startswith("VmRSS:")
return int(line.split()[1]) / 1024.0
:except Exception
pass
return None

```

```

:def alloc_block(mb)
 """ساخت بلوک و لمس هر صفحه تا فیزیکی تخصیص یابد"""
 size = mb * MB
 b = bytearray(size)
 for i in range(0, size, PAGE):
 b[i] = 1
 return b

():def main
 ap = argparse.ArgumentParser(description="Fill RAM to a target percent (Orange Pi stress)")
 ,ap.add_argument("--target", type=float, default=98.0
 ("درصد هدف استفاده از رم (پیشفرض 98)")=help
 ,ap.add_argument("--reserve-mb", type=int, default=64
 ("حاشیه امن MB که دستخورده بماند (پیشفرض 64)")=help
 ,ap.add_argument("--block-mb", type=int, default=16
 ("اندازه هر بلوک تخصیص (MB) پیشفرض 16")=help
 ,ap.add_argument("--sleep", type=float, default=0.05
 ("مکث بین تخصیص‌ها (ثانیه) برای نرم‌تر شدن فشار")=help
 ,"ap.add_argument("--danger-allow-oom", action="store_true
 ("اجازه بده حاشیه امن صفر شود (ریسکی؛ ممکن است سیستم هنگ کند)")=help
)args = ap.parse_args

():total, avail = meminfo_mb

```

```

if total <= 0
print("Cannot read /proc/meminfo", file=sys.stderr)
(1)sys.exit

محاسبه‌ی هدف MB
target_mb = int(max(0.0, min(100.0, args.target)) * total / 100.0)
reserve_mb = max(0, args.reserve_mb)

if args.danger-allow-oom and args.target >= 100.0
 # بدون حاشیه (ریسک بالا)
 print(f"MemTotal={total} MB, MemAvailable={avail} MB")
 print(f"Target ~{target_mb} MB, Reserve ~{reserve_mb} MB, Block={args.block_mb} MB")
 print("Allocating... (Ctrl+C to free)")

[] = blocks
allocated_mb = 0

try:
 while True
 avail = meminfo_mb -
 # اگر به زیر حاشیه امن رسیدیم، متوقف شو
 if avail <= reserve_mb
 print(f"[STOP] Reached reserve floor: MemAvailable ~{avail} MB")

```

```
break
```

```
اگر به هدف رسیدیم، متوقف شو
if allocated_mb >= target_mb
print(f"[STOP] Reached target allocation: ~{allocated_mb} MB")
break
```

```
میزان بلوک بعدی (از حد باقیمانده بیشتر نشود)
remaining_to_target = max(0, target_mb - allocated_mb)
همچنین مراقب باش از حاشیه امن عبور نکنی
safe_alloc_cap = max(0, avail - reserve_mb)
next_mb = min(args.block_mb, remaining_to_target, safe_alloc_cap)
if next_mb <= 0
print(f"[STOP] No safe space left. MemAvailable ~{avail} MB")
break
```

```
تخصیص واقعی
try
b = alloc_block(next_mb)
blocks.append(b)
allocated_mb += next_mb
except MemoryError
print("[OOM] MemoryError while allocating.")
break
```

```

لحظه‌ای تزارش

()rss = rss_mb

.if rss is not None

print(f"Allocated ~{allocated_mb} MB | VmRSS={rss:.1f} MB | MemAvail~{avail}
MB")

.else

print(f"Allocated ~{allocated_mb} MB | MemAvail~{avail} MB")

time.sleep(args.sleep)

print("Holding memory. Press Ctrl+C to release.")

.while True

(1)time.sleep

.except KeyboardInterrupt

print("\nReleasing memory...")

()blocks.clear

(0.5)time.sleep

()avail2 = meminfo_mb ,_
()rss2 = rss_mb

print(f"Released. VmRSS={rss2:.1f} MB, MemAvailable~{avail2} MB" if rss2 else
"Released.")

.except Exception as e

print(f"Error: {e}", file=sys.stderr)

(1)time.sleep

```

```
:__if __name__ == "__main\n()\nmain
```

توضیح :

```
/#!/usr/bin/env python3
```

شِبنگ: اجرای مستقیم فایل با پایتون ۳.

```
import argparse, time, os, sys
```

کتابخانه‌ها:

برای پارس آرگومان‌های خط فرمان؛ `argparse`

برای مکث؛ `time`

برای خروج/چاپ خطای `sys`

نکته: `os` در این نسخه استفاده نشده و می‌توان حذف کرد.

`PAGE = 4096`

`MB = 1024 * 1024`

ثبت‌ها: اندازه‌ی حافظه را `KB`<sup>۴</sup> فرض کرده (در اغلب لینوکس‌ها همین است) و یک «مگابایت» را به بایت.

خواندن وضعیت حافظه از لینوکس

```

def meminfo_mb():

 """برگرداندن (MemTotal_MB, MemAvailable_MB)"""

 total_kb = avail_kb = 0

 with open("/proc/meminfo") as f:

 for line in f:

 if line.startswith("MemTotal:"):

 total_kb = int(line.split()[1])

 elif line.startswith("MemAvailable:"):

 avail_kb = int(line.split()[1])

 if not avail_kb: # fallback

 with open("/proc/meminfo") as f:

 for line in f:

 if line.startswith("MemFree:"):

 avail_kb = int(line.split()[1])

 break

 return total_kb // 1024, avail_kb // 1024

```

از /proc/meminfo مقدار کل حافظه (MemTotal) و حافظه‌ی قابل استفاده‌ی واقعی (MemAvailable) را می‌خواند.

اگر MemFree نبود، به MemAvailable برمی‌گردد (کیفیت کمتر).

خروجی را به MB برمی‌گرداند.

```
def rss_mb():
```

```
try:
```

```
with open("/proc/self/status") as f:
```

```
 for line in f:
```

```
 if line.startswith("VmRSS:":
```

```
 return int(line.split()[1]) / 1024.0
```

```
 except Exception:
```

```
 pass
```

```
 return None
```

از /proc/self/status مقدار RSS (حافظه‌ی رزیدنت واقعی پروسه) را به MB بازمی‌گرداند تا گزارش دقیق از مصرف خود اسکریپت داشته باشیم.

تخصیص واقعی حافظه (و لمس هر صفحه)

```
def alloc_block(mb):
```

```
 """ساخت بلوک و لمس هر صفحه تا فیزیکی تخصیص یابد"""
```

```
 size = mb * MB
```

```
 b = bytearray(size)
```

```
 for i in range(0, size, PAGE):
```

```
 b[i] = 1
```

```
 return b
```

یک bytearray با اندازه‌ی دقیق موردنظر (مثلاً MB16) می‌سازد.

سپس هر KB<sup>4</sup> یک بایت را می‌نویسد تا صفحه‌ها واقعاً به صورت فیزیکی تخصیص داده شوند (نه فقط رزرو مجازی/تنبل). این باعث می‌شود اثر تست واقعی باشد.

منطق اصلی برنامه

```

def main():

 ap = argparse.ArgumentParser(...)

 ap.add_argument("--target", default=98.0, ...) # درصد هدف استفاده از (...) RAM

 ap.add_argument("--reserve-mb", default=64, ...) # حاشیه امن (...) MB(...

 ap.add_argument("--block-mb", default=16, ...) # اندازه هر بلوک تخصیص (...)

 ap.add_argument("--sleep", default=0.05, ...) # مکث بین تخصیص ها (...)

 ap.add_argument("--danger-allow-oom", action="store_true", ...)

args = ap.parse_args()

```

آرگومان ها:

درصد هدف (مثلاً٪۹۸) target--

مقدار حافظه کنار گذاشته شده تا سیستم نفس بکشد (پیشفرض MB64) reserve-mb--

اندازه هر گام تخصیص (پیشفرض MB16) block-mb--

مکث کوچک بین گامها، برای نرمتر بودن فشار، sleep--

اگر روشن شود و هدف ٪۱۰۰ باشد، حاشیه امن صفر می شود (ریسکی). danger-allow-oom--

```

total, avail = meminfo_mb()

اگر نتوانست /proc/meminfo بخواند، خارج شود ... :if total <= 0

```

target\_mb = int(max(0.0, min(100.0, args.target)) \* total / 100.0)

reserve\_mb = max(0, args.reserve\_mb)

```
if args.danger-allow-oom and args.target >= 100.0:
```

```
 reserve_mb = 0 # بدون حاشیه (ریسک بالا)
```

محاسبه‌ی مقدار MB هدف بر اساس درصد.

اعمال حاشیه‌ی امن؛ اگر حالت «خطرناک» برای ۱۰۰٪ فعال باشد، حاشیه صفر می‌شود.

```
print(f"MemTotal=..., MemAvailable=...")
print(f"Target ~..., Reserve ~..., Block=...")
print("Allocating... (Ctrl+C to free)")
```

چاپ وضعیت اولیه.

```
blocks[] =
```

```
allocated_mb = 0
```

لیستی از بلوک‌های تخصیص یافته؛ تا وقتی در لیست باشند، GC آزادشان نمی‌کند.

حلقه‌ی تخصیص

```
try:
```

```
 while True:
```

```
 avail = meminfo_mb()
```

در هر دور مقدار MemAvailable فعلی را می‌گیرد (زنده/لحظه‌ای).

```
 if avail <= reserve_mb:
```

```
 print("[STOP] Reached reserve floor ..."); break
```

اگر به کف حاشیه‌ی امن رسید، متوقف می‌شود تا سیستم قفل نکند.

```
if allocated_mb >= target_mb:
 print("[STOP] Reached target allocation ..."); break
```

اگر به هدف کل رسید، متوقف می‌شود.

```
remaining_to_target = max(0, target_mb - allocated_mb)
safe_alloc_cap = max(0, avail - reserve_mb)
next_mb = min(args.block_mb, remaining_to_target, safe_alloc_cap)
if next_mb <= 0:
 print("[STOP] No safe space left ..."); break
```

اندازه‌ی بلوک بعدی را تعیین می‌کند تا از هدف و حاشیه عبور نکند.

try:

```
b = alloc_block(next_mb)
blocks.append(b)
allocated_mb += next_mb
except MemoryError:
 print("[OOM] MemoryError while allocating."); break
```

تخصیص واقعی بلوک و افزودن به لیست. اگر MemoryError بیاید، توقف.

```
rss = rss_mb()
if rss is not None:
 print(f"Allocated ~{allocated_mb} MB | VmRSS={rss:.1f} MB | MemAvail~{avail} MB")
```

else:

```
 print(f"Allocated ~{allocated_mb} MB | MemAvail~{avail} MB")
```

```
time.sleep(args.sleep)
```

گزارش زنده: حجم تخصیص داده شده، RSS واقعی پروسه، و MemAvailable. سپس مکث کوتاه.

نگهداشت حافظه تا دستور کاربر

```
print("Holding memory. Press Ctrl+C to release.")
while True:
 time.sleep(1)
```

پس از رسیدن به هدف/کف، حافظه را نگه می‌دارد تا با Ctrl+C آزاد کنید.

آزادسازی تمیز

```
except KeyboardInterrupt:
```

```
 print("\nReleasing memory...")
 blocks.clear()
 time.sleep(0.5)
 avail2 = meminfo_mb()
 rss2 = rss_mb()

 print(f"Released. VmRSS={rss2:.1f} MB, MemAvailable~{avail2} MB" if rss2 else
 "Released.")
```

با قطع توسط کاربر: لیست خالی و رفرنس‌ها حذف می‌شوند → حافظه آزاد می‌شود؛ گزارش نهایی چاپ می‌شود.

تذکر: ممکن است RSS فوراً تا صفر نیاید (به خاطر رفتار allocator/glibc، و page cache)، ولی MemAvailable بالا می‌رود و حافظه برای بقیه قابل استفاده می‌شود.

```
except Exception as e:
```

```
 print(f"Error: {e}", file=sys.stderr)
```

```
time.sleep(1)

if __name__ == "__main__":
 main()
```

خطاهای غیرمنتظره هندل می‌شوند؛ گارد اجرای مستقیم.

#### جمع‌بندی

اسکریپت MemAvailable را لحظه‌ای می‌خواند، بلوک‌های دقیق bytearray می‌سازد و با لمس هر صفحه، حافظه را واقعاً مصرف می‌کند تا به درصد هدف برسد؛ هم‌زمان حاشیه‌ی امن را رعایت می‌کند.

برای ریسک بسیار بالا (واقعاً نزدیک ۱۰۰٪)، target 100 --reserve-mb 0 --danger-allow-oom-- را می‌توان زد—  
ولی ممکن است سیستم هنگ کند؛ احتیاط کنیم

```
Restart
monitor_usage.py:
#!/usr/bin/env python3

import psutil
import time
import os

درصد CPU_THRESHOLD = 99
درصد RAM_THRESHOLD = 99
چند ثانیه پشت سر هم DURATION = 5
```

```

def main():

 cpu_count = psutil.cpu_count()

 high_usage_seconds = 0

 while True:

 مصرف CPU همه هسته‌ها cpu_usage = psutil.cpu_percent(interval=1) #

 مصرف RAM ram_usage = psutil.virtual_memory().percent #

 print(f"CPU: {cpu_usage:.1f}% | RAM: {ram_usage:.1f}%")

 if cpu_usage >= CPU_THRESHOLD or ram_usage >= RAM_THRESHOLD:

 high_usage_seconds += 1

 else:

 high_usage_seconds = 0

 if high_usage_seconds >= DURATION:

 ... مصرف بالا برای ۵ ثانیه → ریاستارت می‌کنم("⚠️"
 print("⚠️"

 os.system("reboot")

 break

if __name__ == "__main__":
 main()

```

توضیح :

```
/#!/usr/bin/env python3
```

شبانگ: اجرای اسکریپت با مفسر پایتون.<sup>۳</sup>.

```
import psutil
import time
import os
```

.RAM برای خواندن مصرف CPU و psutil

برای زمانبندی (اینجا مستقیم استفاده نشده). time

os برای اجرای فرمان سیستم (reboot).

```
CPU_THRESHOLD = 99 # درصد
RAM_THRESHOLD = 99 # درصد
DURATION = 5 # چند ثانیه پشت سر هم
```

آستانه‌ی هشدار RAM و CPU روی ۹۹٪

DURATION=5 یعنی اگر این وضعیت ۵ ثانیه‌ی پیاپی ادامه داشت، عمل ریاستارت انجام شود.

```
def main():

 cpu_count = psutil.cpu_count()
```

```
high_usage_seconds = 0
```

تابع اصلی.

تعداد هسته‌ها را برمی‌گرداند (در ادامه‌ی کد استفاده نشده).

شمارنده‌ی ثانیه‌های متوالی استفاده‌ی بالا: ابتدا صفر.

```
while True:
```

حلقه‌ی بی‌نهایت برای پایش پیوسته.

```
cpu_usage = psutil.cpu_percent(interval=1) # مصرف CPU همه هسته‌ها
```

```
ram_usage = psutil.virtual_memory().percent # RAM مصرف
```

(کل سیستم) را برمی‌گرداند. CPU یک ثانیه صبر می‌کند و سپس میانگین مصرف (1

درصد استفاده‌ی RAM را می‌دهد.

```
print(f"CPU: {cpu_usage:.1f}% | RAM: {ram_usage:.1f}%)
```

چاپ وضعیت لحظه‌ای با یک رقم اعشار.

```
if cpu_usage >= CPU_THRESHOLD or ram_usage >= RAM_THRESHOLD:
```

```
 high_usage_seconds += 1
```

```
else:
```

```
 high_usage_seconds = 0
```

اگر یکی از دو مصرف (CPU یا RAM) از آستانه رد شود: شمارنده‌ی «ثانیه‌های پرصرف» یکی زیاد می‌شود.

در غیر این صورت، شمارنده ریست می‌شود (زنگیره‌ی «پشت‌سرِهم» قطع می‌گردد).

```
if high_usage_seconds >= DURATION:
```

print("⚠️ مصرف بالا برای ۵ ثانیه → ریاستارت می‌کنم...")

```
os.system("reboot")
```

```
break
```

اگر تعداد ثانیه‌های متوالی مصرف بالا به DURATION برسد/بگذرد:

پیام هشدار چاپ می‌کند.

با (os.system("reboot")) دستور ریاستارت سیستم را اجرا می‌کند.

سپس از حلقه خارج می‌شود.

```
if __name__ == "__main__":
```

```
 main()
```

اگر فایل به صورت برنامه‌ی اصلی اجرا شده باشد، main() را صدا می‌زند.

خلاصه‌ی عملکرد

این اسکریپت هر ثانیه مصرف CPU و RAM را می‌سنجد؛ اگر حداقل یکی از آن‌ها به مدت ۵ ثانیه‌ی پیاپی از ۹۹٪ عبور کند، پیام هشدار چاپ می‌کند و سپس سیستم را ریاستارت می‌کند.

Restart

Stress.py:

```
#!/usr/bin/env python3

import multiprocessing

def burn():

 while True:

 pass

if __name__ == "__main__":

 cores = multiprocessing.cpu_count()

 workers = []

 for _ in range(cores):

 p = multiprocessing.Process(target=burn)

 p.start()

 workers.append(p)

for p in workers:

 p.join()
```

: توضیح

```
/#!/usr/bin/env python3
```

شیانگ: اسکریپت با پایتون ۳ اجرا شود (وقتی به صورت احرابی ران کنی).

```
import multiprocessing
```

ماژول استاندارد برای ساخت پردازه‌های جدا (نه نخها). هر پردازه GIL خودش را دارد.

```
def burn():
```

```
 while True:
```

```
 pass
```

تابعی که یک حلقه‌ی بینهایت اجرا می‌کند.

.pass خودش کاری نمی‌کند، اما اجرای دائمی حلقه باعث می‌شود پردازه ۱۰۰٪ یک هسته را درگیر کند (tight loop).

```
if __name__ == "__main__":
```

گارد لازم مخصوصاً در ویندوز تا از spawn شدن بازگشتی جلوگیری شود.

```
cores = multiprocessing.cpu_count()
```

تعداد CPU منطقی سیستم (Logical cores/Hyperthreads) را می‌گیرد. ممکن است بیشتر از هسته‌های فیزیکی باشد.

```
workers[] =
```

آرایه برای نگهداشتن هندل پردازه‌ها.

```
for _ in range(cores):
```

```
 p = multiprocessing.Process(target=burn)
```

```
 p.start()
```

```
 workers.append(p)
```

به تعداد هسته‌ها، پردازه‌ی جدید می‌سازد که هر کدام burn() را اجرا می‌کنند.

هر پردازه جداگانه اجرا می‌شود؛ بنابراین محدودیت GIL دور زده می‌شود و واقعاً CPU را می‌سوزاند.

for p in workers:

p.join()

منتظر می‌ماند تا هر پردازه تمام شود.

چون `burn()` بی‌نهایت است، هیچ وقت برنمی‌گردد و اینجا برای همیشه بلاک می‌ماند مگر اسکریپت/پردازه‌ها را دستی متوقف کنی.

#### نکات مهم/احتیاط

این برنامه عمداً همه‌ی هسته‌ها را ۱۰۰٪ درگیر می‌کند؛ می‌تواند باعث داغشدن، فن پرسرعت، و لگ سیستم شود. روی دیوايس‌های کوچک (Pi/OrangePi) با احتیاط.

اگر در کانتینر با `CPU quota (cgroups)` محدود نشده باشی، کل میزبان را درگیر می‌کند.

پایان دادن: چون `join()` بی‌نهایت بلاک است، عملیاتی برای توقف ندارد. معمولاً با `Ctrl+C` یا `Kill` باید بزنیش.

#### خلاصه‌ی عملکرد

این اسکریپت به تعداد هسته‌های منطقی سیستم پردازه‌های جدا راه می‌اندازد و هر کدام با یک حلقه‌ی بی‌نهایت، ۱۰۰٪ یک هسته را مصرف می‌کند؛ در نتیجه کل CPU سیستم تا سقف مصرف می‌شود و برنامه هرگز تمام نمی‌شود مگر آن را دستی متوقف کنی.

Windows

Subscribe\_GUI\_MQTT

Subscriber.py:

```
#!/usr/bin/env python3
```

```

-*- coding: utf-8 -*-

from __future__ import annotations

import json, sys

from dataclasses import dataclass

from datetime import datetime, timedelta, timezone

from typing import Optional, Deque

from collections import deque

import paho.mqtt.client as mqtt

from PySide6.QtCore import Qt, QTimer, QObject, Signal, QSize, QRectF

from PySide6.QtGui import QPainter, QFont, QColor, QPen

from PySide6.QtWidgets import (
 QApplication, QWidget, QLabel, QVBoxLayout, QHBoxLayout, QFrame, QGridLayout,
 QPushButton, QPlainTextEdit, QSizePolicy, QDialog
)

===== CONFIG =====

BROKER = "192.168.100.102"

PORT = 8883

USER = "Ali"

PASS = "EmbeddedProject"

USE_TLS = True

```

```
TLS_CA_CERT = r"C:\Users\Ali\Desktop\New folder
(10)\OrangePi\Mosquitto\mosquitto_certs\server.pem"

CLIENT_CERT = r"C:\Users\Ali\Desktop\New folder
(10)\OrangePi\Mosquitto\mosquitto_certs\server.crt"

CLIENT_KEY = r"C:\Users\Ali\Desktop\New folder
(10)\OrangePi\Mosquitto\mosquitto_certs\server.key"

TLS_INSECURE = True
```

TOPIC = "iot/pi-99/metrics"

STREAM\_TIMEOUT\_SEC = 5

# لایگ

LOG\_BUFFER\_MAX = 500 # چند خط آخر را نگه داریم  
LOG\_TO\_STDOUT = True # چاپ در کنسول/ترمینال VSCode

# ===== DATA =====

@dataclass

class Metrics:

```
ts: Optional[str] = None
cpu_percent: Optional[float] = None
ram_percent: Optional[float] = None
cpu_temp_c: Optional[float] = None
ip_addr: Optional[str] = None
wifi_rssi: Optional[float] = None
host: Optional[str] = None
```

```

def parse_ts(ts: Optional[str]) -> Optional[datetime]:
 if not ts: return None

 try:
 dt = datetime.fromisoformat(ts.replace("Z", "+00:00"))

 if dt.tzinfo is None: dt = dt.replace(tzinfo=timezone.utc)

 return dt.astimezone(timezone.utc)
 except Exception:
 return None

===== MQTT =====
class MqttWorker(QObject):
 metrics_received = Signal(object, object) # (Metrics, payload dict)
 connection_state = Signal(bool)
 tls_state = Signal(bool)

 def __init__(self):
 super().__init__()
 self.client = mqtt.Client()
 self.client.username_pw_set(USER, PASS)

 if USE_TLS:
 try:
 self.client.tls_set(ca_certs=TLS_CA_CERT) if TLS_CA_CERT else self.client.tls_set()
 if TLS_INSECURE: self.client.tls_insecure_set(True)
 self.tls_state.emit(True)
 except Exception:

```

```

 self.tls_state.emit(False)

else:
 self.tls_state.emit(False)

self.client.on_connect = self._on_connect
self.client.on_disconnect = self._on_disconnect
self.client.on_message = self._on_message

def connect(self):
 try:
 self.client.connect(BROKER, PORT, 60)
 self.client.loop_start()
 except Exception:
 self.connection_state.emit(False)

def _on_connect(self, c, u, f, rc):
 ok = (rc == 0); self.connection_state.emit(ok)
 if ok: c.subscribe(TOPIC)

def _on_disconnect(self, c, u, rc):
 self.connection_state.emit(False)

def _on_message(self, c, u, msg):
 raw = msg.payload.decode("utf-8", errors="ignore")
 try: payload = json.loads(raw)
 except Exception: payload = {}

```

```

m = Metrics(
 ts=payload.get("ts"),
 cpu_percent=payload.get("cpu_percent"),
 ram_percent=payload.get("ram_percent"),
 cpu_temp_c=payload.get("cpu_temp_c"),
 ip_addr=payload.get("ip_addr"),
 wifi_rssi=payload.get("wifi_rssi"),
 host=payload.get("host"),
)
self.metrics_received.emit(m, payload)

===== UI PARTS =====
class GlassCard(QFrame):
 def __init__(self, title: str, parent=None):
 super().__init__(parent)
 self.setObjectName("GlassCard")
 self.setFrameShape(QFrame.StyledPanel)
 self.setContentsMargins(18,18,18,18)
 lay = QVBoxLayout(self); lay.setSpacing(10)
 t = QLabel(title); t.setObjectName("CardTitle")
 lay.addWidget(t)
 self.layout = lay

class CircularGauge(QWidget):
 def __init__(self, units: str = "%", parent=None):
 super().__init__(parent)

```

```

 self._value, self._units = 0.0, units
 self.setFixedSize(QSize(180, 180))

def setValue(self, v: float):
 v = 0.0 if v is None else float(v)
 v = max(0.0, min(100.0, v))
 if abs(self._value - v) > 1e-3: self._value = v; self.update()

def paintEvent(self, _):
 size = min(self.width(), self.height())
 margin = 24
 rect = QRectF(margin, margin, size-2*margin, size-2*margin)
 p = QPainter(self); p.setRenderHint(QPainter.Antialiasing)
 pen_bg = QPen(QColor(255,255,255,38), 16, Qt.SolidLine, Qt.RoundCap)
 p.setPen(pen_bg); p.drawArc(rect, 135*16, 270*16)
 hue = int((100 - self._value) * 1.2); hue = max(0, min(120, hue))
 pen_val = QPen(QColor.fromHsv(hue,255,230), 16, Qt.SolidLine, Qt.RoundCap)
 p.setPen(pen_val); span = int((self._value/100.0)*270*16); p.drawArc(rect, 135*16, -span)
 p.setPen(QColor(240,240,245)); p.setFont(QFont("Segoe UI", 26, QFont.DemiBold))
 p.drawText(self.rect(), Qt.AlignCenter, f"{self._value:.0f}{self._units}")

class TempPill(QWidget):
 def __init__(self, parent=None):
 super().__init__(parent)
 self._value = 0.0

```

```

self.setMinimumHeight(110)

def setValue(self, v: float):
 v = 0.0 if v is None else float(v)
 v = max(-20.0, min(110.0, v))
 if abs(self._value - v) > 1e-3: self._value = v; self.update()

def paintEvent(self, _):
 p = QPainter(self); p.setRenderHint(QPainter.Antialiasing)
 w, h = self.width(), self.height()
 m = 22; r = (h - 2*m) / 2
 bar = QRectF(m, m, w-2*m, h-2*m)
 p.setPen(Qt.NoPen); p.setBrush(QColor(255,255,255,30)); p.drawRoundedRect(bar, r,
r)
 pct = max(0.0, min(1.0, self._value/100.0))
 hue = int((100 - (pct*100))*1.2); hue = max(0, min(120, hue))
 fill = QRectF(bar.left(), bar.top(), bar.width()*pct, bar.height())
 p.setBrush(QColor.fromHsv(hue,255,230)); p.drawRoundedRect(fill, r, r)
 p.setPen(QColor(240,240,245)); p.setFont(QFont("Segoe UI", 20, QFont.DemiBold))
 p.drawText(bar, Qt.AlignCenter, f"{self._value:.1f}°C")

----- Log Window -----
class LogWindow(QDialog):
 def __init__(self, parent=None):
 super().__init__(parent)
 MQTT" آخرین پیام‌های" self.setWindowTitle("
```

```

self.resize(820, 420)

lay = QVBoxLayout(self); lay.setContentsMargins(12,12,12,12); lay.setSpacing(8)

self.view = QPlainTextEdit()

self.view.setReadOnly(True)

self.view.setObjectName("Console")

self.view.setSizePolicy(QSizePolicy.Preferred, QSizePolicy.Expanding)

bt�s = QHBoxLayout()

پاکسازی() self.btn_clear = QPushButton("

بستن" self.btn_close = QPushButton("

self.btn_clear.clicked.connect(self.view.clear)

self.btn_close.clicked.connect(self.close)

bt�s.addStretch(1); bt�s.addWidget(self.btn_clear); bt�s.addWidget(self.btn_close)

lay.addWidget(self.view)

lay.addLayout(bt�s)

def append(self, line: str):

 self.view.appendPlainText(line)

 sb = self.view.verticalScrollBar()

 sb.setValue(sb.maximum())

====== MAIN WINDOW =====

class MainWindow(QWidget):

 def __init__(self):

```

```

super().__init__()

self.setWindowTitle("Orange Pi Live Metrics")
self.setMinimumSize(980, 660)
self._last_msg_at: Optional[datetime] = None

برای پنجره جدا (# Log buffer (
self._log_buf: Deque[str] = deque(maxlen=LOG_BUFFER_MAX)
self._log_win: Optional[LogWindow] = None

self._build_ui()
self._apply_style()

self.mqtt = MqttWorker()
self.mqtt.metrics_received.connect(self.on_metrics)
self.mqtt.connection_state.connect(self.on_connection)
self.mqtt.tls_state.connect(self.on_tls_state)
self.mqtt.connect()

self.clock = QTimer(self); self.clock.timeout.connect(self.update_clock);
self.clock.start(1000); self.update_clock()

self.watch = QTimer(self); self.watch.timeout.connect(self.update_stream_badge);
self.watch.start(1000)

def _build_ui(self):
 root = QVBoxLayout(self); root.setContentsMargins(22,22,22,22); root.setSpacing(16)
 # Header

```

```

header = QHBoxLayout()

self.title_lbl = QLabel("Orange Pi • Live Monitor");
self.title_lbl.setObjectName("AppTitle")

badges = QHBoxLayout(); badges.setSpacing(8)

self.tls_lbl = QLabel("🔒 TLS: Off"); self.tls_lbl.setObjectName("Badge")

self.status_lbl = QLabel("Disconnected"); self.status_lbl.setObjectName("Badge")

self.stream_lbl = QLabel("Stream: No data"); self.stream_lbl.setObjectName("Badge")

badges.addWidget(self.tls_lbl); badges.addWidget(self.status_lbl);
badges.addWidget(self.stream_lbl)

header.addWidget(self.title_lbl, 1, Qt.AlignLeft|Qt.AlignVCenter);
header.addLayout(badges, 0)

root.setLayout(header)

```

# Grid

```
grid = QGridLayout(); grid.setHorizontalSpacing(18); grid.setVerticalSpacing(18)
```

```

temp_card = GlassCard("Temperature")

temp_v = QVBoxLayout(); temp_v.setSpacing(6)

self.temp_label = QLabel("CPU Temperature");
self.temp_label.setObjectName("InfoLine");
self.temp_label.setAlignment(Qt.AlignHCenter)

self.temp_pill = TempPill(); temp_v.addWidget(self.temp_label);
temp_v.addWidget(self.temp_pill)

temp_card.layout.addLayout(temp_v)

```

```
gauges_card = GlassCard("System Load")
```

```
g_h = QHBoxLayout(); g_h.setSpacing(36); g_h.setContentsMargins(0,8,0,0)
```

```
cpu_box = QVBoxLayout(); cpu_box.setSpacing(6)
```

```

 self.cpu_gauge = CircularGauge(); cpu_cap = QLabel("CPU Usage");
 cpu_cap.setAlignment(Qt.AlignHCenter); cpu_cap.setObjectName("InfoLine")

 cpu_box.addWidget(self.cpu_gauge, alignment=Qt.AlignHCenter);
 cpu_box.addWidget(cpu_cap)

 ram_box = QVBoxLayout(); ram_box.setSpacing(6)

 self.ram_gauge = CircularGauge(); ram_cap = QLabel("RAM Usage");
 ram_cap.setAlignment(Qt.AlignHCenter); ram_cap.setObjectName("InfoLine")

 ram_box.addWidget(self.ram_gauge, alignment=Qt.AlignHCenter);
 ram_box.addWidget(ram_cap)

 g_h.addLayout(cpu_box,1); g_h.addLayout(ram_box,1);
 gauges_card.layout.addLayout(g_h)

info_card = GlassCard("Device")

self.host_lbl = QLabel("Host: —"); self.ip_lbl = QLabel("IP: —")

self.ts_lbl = QLabel("Last Update: —"); self.now_lbl = QLabel("Local Time: —")

for w in (self.host_lbl, self.ip_lbl, self.ts_lbl, self.now_lbl):
 w.setObjectName("InfoLine"); info_card.layout.addWidget(w)

grid.addWidget(temp_card,0,0,1,2); grid.addWidget(gauges_card,1,0,1,2);
grid.addWidget(info_card,0,2,2,1)

root.setLayout(grid, 1)

Footer + “Messages” button

footer = QHBoxLayout()

self.small_hint = QLabel(f"Broker: {BROKER}:{PORT} • Topic: {TOPIC} • " + ("TLS
Enabled" if USE_TLS else "No TLS"))

self.small_hint.setObjectName("FooterHint")

footer.addWidget(self.small_hint, 1, Qt.AlignLeft)

```

```

") پیام‌ها (پنجره جدا)

self.btn_open_logs = QPushButton("

self.btn_open_logs.clicked.connect(self.open_logs_window)

footer.addWidget(self.btn_open_logs, 0, Qt.AlignRight)

root.setLayout(footer)

def _apply_style(self):

 self.setStyleSheet(""""

 QWidget { color: #EDEFF4; }

 #AppTitle { font: 700 22px "Segoe UI"; letter-spacing: 0.3px; }

 #Badge { font: 600 14px "Segoe UI"; color: #BFC6D6; padding: 4px 10px; border-
radius: 10px; background: rgba(255,255,255,0.08); }

 #GlassCard { background: rgba(255,255,255,0.06); border: 1px solid
rgba(255,255,255,0.14); border-radius: 20px; }

 #CardTitle { font: 600 14px "Segoe UI"; color: #C9D2E3; margin-bottom: 4px; }

 #InfoLine { font: 500 13px "Segoe UI"; color: #DCE3F2; padding: 6px 0; }

 #FooterHint { font: 11px "Segoe UI"; color: #A8B0C1; }

 #Console {

 background: rgba(0,0,0,0.35);

 border: 1px solid rgba(255,255,255,0.18);

 border-radius: 10px;

 font: 12px "Consolas","Cascadia Mono","Courier New",monospace;

 color: #DCE3F2;

 }

"""
)

```

```

pal = self.palette(); pal.setColor(self.backgroundRole(), QColor(18, 24, 38))
self.setAutoFillBackground(True); self.setPalette(pal)

----- log helpers -----
def log_append(self, line: str):
 self._log_buf.append(line)
 if self._log_win and self._log_win.isVisible():
 self._log_win.append(line)
 if LOG_TO_STDOUT:
 try:
 ترمinal print(line) # VSCode/
 sys.stdout.flush()
 except Exception:
 pass

def open_logs_window(self):
 if not self._log_win:
 self._log_win = LogWindow(self)
 self._log_win.show()
 self._log_win.raise_()
 self._log_win.activateWindow()
 پرکردن بافر موجود #
 self._log_win.view.clear()
 self._log_win.view.appendPlainText("\n".join(self._log_buf))
 sb = self._log_win.view.verticalScrollBar()
 sb.setValue(sb.maximum())

```

```

----- slots -----

def on_tls_state(self, enabled: bool):

 self.tls_lbl.setText("🔒 TLS: On" if enabled else "🔓 TLS: Off")

 self.tls_lbl.setStyleSheet("#Badge { color: %s; padding: 4px 10px; border-radius: 10px; background: rgba(255,255,255,0.08); }"

 % ("#79F2A0" if enabled else "#FFB86C"))

def on_connection(self, ok: bool):

 self.status_lbl.setText("Connected" if ok else "Disconnected")

 self.status_lbl.setStyleSheet("#Badge { color: %s; padding: 4px 10px; border-radius: 10px; background: rgba(255,255,255,0.08); }"

 % ("#79F2A0" if ok else "#FF8F8F"))

if not ok:

 self._last_msg_at = None

 self.update_stream_badge()

def on_metrics(self, m: Metrics, payload: dict):

 self._last_msg_at = datetime.now()

 # UI

 if m.cpu_percent is not None: self.cpu_gauge.setValue(float(m.cpu_percent))

 if m.ram_percent is not None: self.ram_gauge.setValue(float(m.ram_percent))

 if m.cpu_temp_c is not None: self.temp_pill.setValue(float(m.cpu_temp_c))

 self.host_lbl.setText(f"Host: {m.host or '—'}")

```

```

 self.ip_lbl.setText(f"IP: {m.ip_addr or '—'}")

 dt_utc = parse_ts(m.ts)

 self.ts_lbl.setText("Last Update: " + (dt_utc.astimezone().strftime("%Y-%m-%d
%H:%M:%S %Z") if dt_utc else "—"))

 self.stream_lbl.setText("Stream: Live")
 self.stream_lbl.setStyleSheet("#Badge { color: #79F2A0; padding: 4px 10px; border-
radius: 10px; background: rgba(121,242,160,0.12); }")

=== LOG ===

خط خلاصه‌ی انسانی:
recv_local = datetime.now().strftime("%H:%M:%S")

ts_disp = dt_utc.astimezone().strftime("%Y-%m-%d %H:%M:%S") if dt_utc else "-"

human = f"[{recv_local}] ts={ts_disp} host={payload.get('host')}
cpu={payload.get('cpu_percent')} ram={payload.get('ram_percent')}
temp={payload.get('cpu_temp_c')}"

self.log_append(human)

/grep): JSON تک‌خطی (برای پردازش: #

try:

 iso_now = datetime.now().astimezone().isoformat(timespec="seconds")

 jline = json.dumps({"recv_at": iso_now, **payload}, ensure_ascii=False)

 self.log_append(jline)

except Exception:

 pass

```

```

def update_clock(self):
 self.now_lbl.setText("Local Time: " + datetime.now().strftime("%Y-%m-%d %H:%M:%S"))

def update_stream_badge(self):
 if self._last_msg_at is None or datetime.now() - self._last_msg_at >
timedelta(seconds=STREAM_TIMEOUT_SEC):
 self.stream_lbl.setText("Stream: No data")
 self.stream_lbl.setStyleSheet("#Badge { color: #FFB86C; padding: 4px 10px; border-radius: 10px; background: rgba(255,184,108,0.12); }")

def main():
 app = QApplication(sys.argv)
 w = MainWindow(); w.show()
 sys.exit(app.exec())

if __name__ == "__main__":
 main()

```

توضیح :

## سربرگ و واردکردن مازول‌ها

```

#!/usr/bin/env python3
-*- coding: utf-8 -*-
from __future__ import annotations

```

- شبانگ اجرای پایتون.<sup>۳</sup>

- تعیین کدپیچ فایل به UTF-8

فعال کردن **Annotations** آینده (تایپ‌ها قابل ارجاع به خود کلاس‌ها). •

```
import json, sys

from dataclasses import dataclass

from datetime import datetime, timedelta, timezone

from typing import Optional, Deque

from collections import deque

import paho.mqtt.client as mqtt
```

ماژول‌های استاندارد برای JSON، زمان، تایپ‌هیئت، صفت حلقوی؛ و کلاینت **MQTT** (paho-mqtt). •

```
from PySide6.QtCore import Qt, QTimer, QObject, Signal, QSize, QRectF

from PySide6.QtGui import QPainter, QFont, QColor, QPen

from PySide6.QtWidgets import (

 QApplication, QWidget, QLabel, QVBoxLayout, QHBoxLayout, QFrame, QGridLayout,

 QPushButton, QPlainTextEdit, QSizePolicy, QDialog

)
```

وارد کردن کلاس‌های UI برای Qt، رنگ‌آمیزی و سیگنال اسلات‌ها. •

## پیکربندی اتصال **MQTT** و لای

```
BROKER = "192.168.100.102"
```

```
PORT = 8883
```

```
USER = "Ali"
```

```
PASS = "EmbeddedProject"
```

آدرس بروکر، پورت، نام کاربری و گذر واژه. •

```
USE_TLS = True
```

```
TLS_CA_CERT = r"..\server.pem"
```

```
CLIENT_CERT = r"..\server.crt"
CLIENT_KEY = r"..\server.key"
TLS_INSECURE = True
```

- فعال سازی TLS و مسیر گواهی ها؛ TLS\_INSECURE=True

```
TOPIC = "iot/pi-99/metrics"
```

```
STREAM_TIMEOUT_SEC = 5
```

- می شود «No data» برای متریک ها؛ اگر طی ۵ ثانیه پیام نرسد MQTT تاپیک.

```
LOG_BUFFER_MAX = 500
```

```
LOG_TO_STDOUT = True
```

- حداکثر ۵۰۰ خط لاغ در بافر داخلی؛ هم زمان چاپ در ترمینال.

## مدل داده های متریک ها و تبدیل زمان

```
@dataclass
```

```
class Metrics:
```

```
 ts: Optional[str] = None

 cpu_percent: Optional[float] = None

 ram_percent: Optional[float] = None

 cpu_temp_c: Optional[float] = None

 ip_addr: Optional[str] = None

 wifi_rssi: Optional[float] = None

 host: Optional[str] = None
```

- ساختار داده های متریک ها (همگی اختیاری).

```
def parse_ts(ts: Optional[str]) -> Optional[datetime]:
```

```
 ...
```

- ورودی ISO8601 ممکن است Z داشته باشد را به datetime با منطقه زمانی UTC تبدیل می کند.

## کارگر MQTT QObject با سیگنال‌ها

```
class MqttWorker(QObject):
 metrics_received = Signal(object, object) # (Metrics, payload dict)
 connection_state = Signal(bool)
 tls_state = Signal(bool)

 سه سیگنال: دریافت متریک‌ها، وضعیت اتصال، وضعیت TLS •

def __init__(self):
 self.client = mqtt.Client()
 self.client.username_pw_set(USER, PASS)

 if USE_TLS:
 ...
 self.client.tls_set(ca_certs=TLS_CA_CERT) if TLS_CA_CERT else self.client.tls_set()
 if TLS_INSECURE: self.client.tls_insecure_set(True)
 self.tls_state.emit(True/False)

 ساخت کلاینت MQTT، سတ کردن احراز هویت و TLS (در صورت فعال بودن) و انتشار سیگنال وضعیت TLS •

 self.client.on_connect = self._on_connect
 self.client.on_disconnect = self._on_disconnect
 self.client.on_message = self._on_message

 وصل کردن callback‌ها. •

def connect(self):
 self.client.connect(BROKER, PORT, 60)
 self.client.loop_start()

 اتصال به بروکر و شروع حلقه شبکه در Thread داخلی. •
```

```

def _on_connect(self, c, u, f, rc):
 ok = (rc == 0); self.connection_state.emit(ok)
 if ok: c.subscribe(TOPIC)

 در اتصال موفق، سیگنال Connected و سابسکرایب تاپیک. •

def _on_disconnect(self, c, u, rc):
 self.connection_state.emit(False)

 قطع شدن اتصال. •

def _on_message(self, c, u, msg):
 raw = msg.payload.decode("utf-8", errors="ignore")
 try: payload = json.loads(raw)
 except Exception: payload = {}
 m = Metrics(...); self.metrics_received.emit(m, payload)

 دیکد پیام، JSON parse، ساخت آبجکت Metrics و انتشار سیگنال. •

```

ویجت‌های UI سفارشی (کارت شیشه‌ای، گیج دایره‌ای، نوار دما)

```

class GlassCard(QFrame):
 ...
 ...
 کارت با عنوان؛ Layout بـ QFrame عمودی. •

class CircularGauge(QWidget):
 ...
 def setValue(self, v: float): ...; self.update()
 def paintEvent(self, _):
 # رسم کمان پس زمینه، کمان مقدار با گرادیان رنگ بر اساس درصد،
 # نوشتن عدد وسط (مثلًا "65%")
 یک گیج حلقه‌ای برای CPU/RAM ۱۰۰٪--. •

```

```

class TempPill(QWidget):
 ...
 def paintEvent(self, _):
 # رسم کپسولی افقی؛ طول پرشده متناسب با دما؛
 # رنگ از سبز→قرمز بر اساس مقدار؛ برچسب "XX.X°C"
 • نوار قرص‌مانند برای نمایش دمای CPU

```

### پنجره‌ی لاغ جدا

```

class LogWindow(QDialog):
 ...
 self.view = QPlainTextEdit(); self.view.setReadOnly(True)
 self.btn_clear.clicked.connect(self.view.clear)
 self.btn_close.clicked.connect(self.close)
 def append(self, line: str): ... # افزودن خط و اسکرول به انتهای
 # نمایش آخرین پیام‌های MQTT در یک دیالوگ جدا.

```

### پنجره‌ی اصلی برنامه

```

class MainWindow(QWidget):
 def __init__(self):
 ...
 self._last_msg_at: Optional[datetime] = None
 self._log_buf: Deque[str] = deque(maxlen=LOG_BUFFER_MAX)
 self._log_win: Optional[LogWindow] = None
 • نگهداشتن زمان آخرین پیام برای تشخیص «No data»، و بافر لاغ‌ها.
 self._build_ui()

```

```
self._apply_style()
```

- ساخت کل UI و اعمال استایل.

```
self.mqtt = MqttWorker()
```

```
self.mqtt.metrics_received.connect(self.on_metrics)
```

```
self.mqtt.connection_state.connect(self.on_connection)
```

```
self.mqtt.tls_state.connect(self.on_tls_state)
```

```
self.mqtt.connect()
```

- راهاندازی MQTT و اتصال سیگنال‌ها.

```
self.clock = QTimer(...).start(1000)
```

```
self.watch = QTimer(...).start(1000)
```

- تایمر ساعت (به روزرسانی زمان محلی هر ثانیه) و تایمر پایش جریان داده (Stream).

### UI (\_build\_ui)

- لایه‌ی ریشه، هدر با **Badge**‌های TLS/Connection/Stream.

- سه کارت:

نوار دما **Temperature**



دو گیج **System Load**



دستگاه (Host/IP/Last Update/Local Time)



- فوتر: متن بروکر و دکمه‌ی «پیام‌ها» (پنجره جدا).

### استایل (\_apply\_style)

- تیره، فونت‌ها، کارت شیشه‌ای، **Badge**‌ها و کنسول QSS.

### کمک‌کننده‌های لاغ

```
def log_append(self, line: str):
```

```
 self._log_buf.append(line)
```

```

if self._log_win and self._log_win.isVisible(): self._log_win.append(line)
if LOG_TO_STDOUT: print(line); sys.stdout.flush()

• افزودن به بافر، درج در پنجره‌ی باز، و چاپ در ترمینال.

def open_logs_window(self):
 if not self._log_win: self._log_win = LogWindow(self)
 self._log_win.show(); ... ; # پرکردن بافر موجود
 بازکردن/فروکوس دادن پنجره‌ی لاغ و پرکردن با لاغ‌های قبلی.

```

#### اسلات‌ها (واکنش به سیگنال‌ها)

```

def on_tls_state(self, enabled: bool):
 self.tls_lbl.setText("🔒 TLS: On/Off")
 self.tls_lbl.setStyleSheet("#Badge { color: ... }")

 به روزرسانی Badge TLS. •

```

```

def on_connection(self, ok: bool):
 self.status_lbl.setText("Connected/Disconnected"); ...
 if not ok:
 self._last_msg_at = None
 self.update_stream_badge()
 به روزرسانی Badge اتصال؛ قطع شدن → ریست وضعیت. •

```

```
def on_metrics(self, m: Metrics, payload: dict):
```

```
 self._last_msg_at = datetime.now()
```

```
به روزرسانی ویجت‌ها
```

```
 self.cpu_gauge.setValue(m.cpu_percent)
```

```
 self.ram_gauge.setValue(m.ram_percent)
```

```

self.temp_pill.setValue(m.cpu_temp_c)

self.host_lbl.setText(f"Host: {m.host or '—'}")
self.ip_lbl.setText(f"IP: {m.ip_addr or '—'}")

dt_utc = parse_ts(m.ts)
self.ts_lbl.setText("Last Update: " + (dt_utc.astimezone().strftime(...) if dt_utc else "—"))

```

```

Stream badge → Live (سبز)
self.stream_lbl.setText("Stream: Live")
self.stream_lbl.setStyleSheet("#Badge { color: #79F2A0; ... }")

```

```

لگ انسانی تک خط
recv_local = datetime.now().strftime("%H:%M:%S")
ts_disp = dt_utc.astimezone().strftime("%Y-%m-%d %H:%M:%S") if dt_utc else "-"
human = f"[{recv_local}] ts={ts_disp} host=... cpu=... ram=... temp=..."
self.log_append(human)

```

# لگ JSON تک خط

```

iso_now = datetime.now().astimezone().isoformat(timespec="seconds")
jline = json.dumps({"recv_at": iso_now, **payload}, ensure_ascii=False)
self.log_append(jline)

```

• دریافت متریک → بروزرسانی + UI نوشتن دو نوع لگ.

```

def update_clock(self):
 self.now_lbl.setText("Local Time: " + datetime.now().strftime(...))

```

- هر ثانیه زمان محلی را به روزرسانی می‌کند.

```
def update_stream_badge(self):
 if self._last_msg_at is None or datetime.now() - self._last_msg_at >
timedelta(seconds=STREAM_TIMEOUT_SEC):
 self.stream_lbl.setText("Stream: No data")
 self.stream_lbl.setStyleSheet("#Badge { color: #FFB86C; ... }")
 «No data» اگر پیام اخیر قدیمی یا هنوز چیزی نامده باشد Badge نارنجی.
```

## راهاندازی برنامه

```
def main():
 app = QApplication(sys.argv)
 w = MainWindow(); w.show()
 sys.exit(app.exec())
if __name__ == "__main__":
 main() ساخت برنامه‌ی Qt، نمایش پنجره اصلی و اجرای حلقه‌ی رویداد.
```

## خلاصه‌ی عملکرد

این برنامه یک مانیتور زنده‌ی **Orange Pi** با PySide6 وصل می‌شود، پیام‌های متريک CPU، RAM، دمای CPU، IP، RSSI، Host را از تاپیک مشخص می‌خواند، ويخت‌های گرافیکی (گیج‌ها و نوار دما) را به روزرسانی می‌کند، وضعیت‌های **TLS/Connection/Stream** را با **Badge** نمایش می‌دهد، و هر پیام را هم به صورت انسانی و هم **JSON** تک خط در لاغ (پنجره‌ی جدا و ترمینال) ثبت می‌کند؛ اگر چند ثانیه داده نیاید، «No data» نشان می‌دهد.

### resource\_monitor\_mqtt.py:

```
#!/usr/bin/env python3
```

```
-*- coding: utf-8 -*-
```

```

import time, json, socket, subprocess, sys
from datetime import datetime, timezone
import psutil
import paho.mqtt.client as mqtt

MQTT = {
 "host": "192.168.100.102", # ← IP بروکر
 "port": 8883,
 "user": "Ali",
 "password": "EmbeddedProject",
 "topic_metrics": "iot/pi-99/metrics",
 "topic_lwt": "iot/pi-99/lwt",
 "keepalive": 30,
 "tls": True,
 "ca": "/opt/iot/python/server.pem", "cert": "/opt/iot/python/server.crt", "key": "/opt/iot/python/server.key",
 "qos": 1
}

```

```

HOSTNAME = socket.gethostname()
_connected = False # وضعیت اتصال فعلی

```

```

def now_iso_utc():
 # 06-09-2025T09:12:34.123Z نمونه:

```

```

 return datetime.now(timezone.utc).isoformat(timespec='milliseconds').replace("+00:00",
"Z")

def get_ip():
 try:
 out = subprocess.check_output(["hostname", "-I"]).decode().strip()
 return out.split()[0] if out else None
 except:
 return None

def get_temp_c():
 for p in ("/sys/class/thermal/thermal_zone0/temp",):
 try:
 v = open(p).read().strip()
 return float(v)/1000.0
 except:
 pass
 return None

def get_wifi_rssi():
 try:
 out = subprocess.check_output(["iwconfig"]).decode(errors="ignore")
 import re
 m = re.search(r"Signal level[=:]\s*(-?\d+)\s*dBm", out)
 return int(m.group(1)) if m else None
 except:

```

```

 return None

--- MQTT Callbacks ---

def on_connect(client, userdata, flags, rc, properties=None):

 global _connected

 _connected = (rc == 0)

 print(f"[MQTT] on_connect rc={rc} connected={_connected}")

 if _connected:

 # اعلام آنلاین (retain)

 client.publish(MQTT["topic_lwt"], json.dumps({"status": "online", "ts": now_iso_utc()}),
qos=1, retain=True)

def on_disconnect(client, userdata, rc, properties=None):

 global _connected

 _connected = False

 print(f"[MQTT] on_disconnect rc={rc} -> will auto-reconnect")

def make_client():

 c = mqtt.Client(client_id=f"{HOSTNAME}-metrics", clean_session=True,
protocol=mqtt.MQTTv311)

 if MQTT["user"] and MQTT["password"]:

 c.username_pw_set(MQTT["user"], MQTT["password"])

 # Last Will

 c.will_set(MQTT["topic_lwt"], json.dumps({"status": "offline", "ts": now_iso_utc()}), qos=1,
retain=True)

 # TLS

```

```

if MQTT["tls"]:

 c.tls_set(ca_certs=MQTT["ca"], certfile=MQTT["cert"], keyfile=MQTT["key"])

 # اگر cert self-signed است و می خواهی verify را سست کنی:
 # c.tls_insecure_set(True)

Callbacks

c.on_connect = on_connect

c.on_disconnect = on_disconnect

برای دیباگ مفید است:
c.enable_logger()

Auto-reconnect backoff

c.reconnect_delay_set(min_delay=1, max_delay=60)

اتصال غیرمسدودکننده + شروع لوپ شبکه در ترد جدا
c.connect_async(MQTT["host"], MQTT["port"], keepalive=MQTT["keepalive"])

c.loop_start()

return c

def main():

 global _connected

 client = make_client()

 # صبر اولیه تا وصل شود (غیربلوک کننده)
 boot_wait_s = 0

```

```

while not _connected and boot_wait_s < 10:
 time.sleep(0.5); boot_wait_s += 0.5

try:
 while True:
 می کند reconnect خودش loop_start اگر قطع است، فقط صبر کن؛ #
 if not _connected:
 time.sleep(1.0)
 continue

 cpu = psutil.cpu_percent(interval=1.0) # 1s block
 ram = psutil.virtual_memory().percent
 temp = get_temp_c()
 ip = get_ip()
 rssi = get_wifi_rssi()

 payload = {
 "ts": now_iso_utc(), # ISO-8601 UTC Z
 "cpu_percent": round(cpu, 1),
 "ram_percent": round(ram, 1),
 "cpu_temp_c": round(temp, 1) if temp is not None else None,
 "ip_addr": ip,
 "wifi_rssi": rssi,
 "host": HOSTNAME
 }

```

```

شد، پیام رد شده/صف نشده است rc != MQTT_ERR_SUCCESS انتشار؛ اگر
info = client.publish(MQTT["topic_metrics"], json.dumps(payload), qos=MQTT["qos"],
retain=False)

شو (تا تایم‌اوت کوتاه) QoS1 اختیاری: منتظر تکمیل
if MQTT["qos"] == 1:

 info.wait_for_publish(timeout=2.0)

فاصله‌ی نمونه‌برداری
time.sleep(2.0) # ۳~ مجموعاً ثانیه با cpu_percent(interval=1)

except KeyboardInterrupt:
 pass

finally:
 try:
 # اعلام آفلاین و توقف تمیز
 client.publish(MQTT["topic_lwt"], json.dumps({"status": "offline", "ts": now_iso_utc()}),
qos=1, retain=True)

 except Exception:
 pass

 # شبکه را متوقف کن
 client.loop_stop()

 client.disconnect()

if __name__ == "__main__":
 main()

```

توضیح :

```
/usr/bin/env python3
```

```
-*- #coding: utf-8 -*-
```

شیانگ برای اجرای پایتون<sup>۳</sup>.

اعلان انکوڈینگ منبع (UTF-8).

```
import time, json, socket, subprocess, sys
from datetime import datetime, timezone
import psutil
import paho.mqtt.client as mqtt
```

+ کتابخانه‌های استاندارد

.MQTT برای آمار سیستم و paho-mqtt برای psutil

```
MQTT} =
" host": "192.168.100.102", # ← IP بروکر
" port": 8883,
" user": "Ali,"
" password": "EmbeddedProject,"
" topic_metrics": "iot/pi-99/metrics,"
" topic_lwt": "iot/pi-99/lwt,"
" keepalive": 30,
" tls": True ,
```

```

 "ca": "/opt/iot/python/server.pem", "cert": "/opt/iot/python/server.crt", "key":

 "/opt/iot/python/server.key",

 "qos": 1

{

```

تنظیمات اتصال: آدرس/پورت بروکر، یوزر/پسورد، تاپیک‌ها، Keepalive و مسیر فایل‌های گواهی.

```
HOSTNAME = socket.gethostname()
```

```
_connected = False # وضعیت اتصال فعلی
```

نام دستگاه برای Client ID و فلگ وضعیت اتصال.

```
def now_iso_utc():
```

```
06-09-2025 # نمونه: T09:12:34.123Z
```

```
return
```

```
datetime.now(timezone.utc).isoformat(timespec='milliseconds').replace("+00:00", "Z")
```

تایم استمپ ISO-8601 به UTC با میلی ثانیه (پسوند Z).

```
def get_ip():
```

```
try:
```

```
 out = subprocess.check_output(["hostname", "-I"]).decode().strip()
```

```
 return out.split()[0] if out else None
```

```
except:
```

```
 return None
```

IP‌های دستگاه را می‌گیرد و اولین IPv4 را بر می‌گرداند.

```

def get_temp_c():

 for p in ("/sys/class/thermal/thermal_zone0/temp",):

 try:

 v = open(p).read().strip()

 return float(v)/1000.0

 except:

 pass

 return None

```

دمای CPU را از sysfs می‌خواند (میلی‌درجه → درجه).

```

def get_wifi_rssi():

 try:

 out = subprocess.check_output(["iwconfig"]).decode(errors="ignore")

 import re

 m = re.search(r"Signal level[=:] \s*(-?\d+)\s*dBm", out)

 return int(m.group(1)) if m else None

 except:

 return None

```

.dBm استخراج می‌کند (RSSI وای‌فای را با پارس خروجی iwconfig

کالبک‌های MQTT

```

def on_connect(client, userdata, flags, rc, properties=None):

 global _connected

 _ connected = (rc == 0)

```

```

print(f"[MQTT] on_connect rc={rc} connected={_connected}")

if _connected:
 # اعلام آنلاین (retain)
 client.publish(MQTT["topic_lwt"], json.dumps({"status": "online", "ts": now_iso_utc()}),
qos=1, retain=True)

```

روی اتصال موفق ( $rc == 0$ ) فلگ را `true` می‌کند و `retain` فعالی را به "online" به صورت LWT می‌فرستد تا همیشه آخرين وضعیت معلوم باشد.

```
def on_disconnect(client, userdata, rc, properties=None):
```

```

global _connected
_connected = False

print(f"[MQTT] on_disconnect rc={rc} -> will auto-reconnect")

```

قطع اتصال → فلگ `false`: پاهو بعداً خودش `reconnect` می‌کند (چون `loop_start` استفاده شده).

```
def make_client():
```

```
c = mqtt.Client(client_id=f"{HOSTNAME}-metrics", clean_session=True,
protocol=mqtt.MQTTv311)
```

ساخت کلاینت با Client ID بر پایه‌ی `hostname`. `clean_session=True` یعنی سشن پایدار نگه‌داری نمی‌شود.

```
if MQTT["user"] and MQTT["password"]:
```

```
c.username_pw_set(MQTT["user"], MQTT["password"])
```

احراز هویت ساده‌ی یوزر/پسورد.

```
Last Will
```

```
c.will_set(MQTT["topic_lwt"], json.dumps({"status":"offline","ts":now_iso_utc()}), qos=1,
retain=True)
```

پیام وصیت (LWT): اگر کلاینت ناگهانی قطع شد، بروکر خودش "offline" را منتشر کند (retain).

## # TLS

```
if MQTT["tls"]:

 c.tls_set(ca_certs=MQTT["ca"], certfile=MQTT["cert"], keyfile=MQTT["key"])

 # اگر cert self-signed است و میخواهی verify را سست کنی: #

 # c.tls_insecure_set(True)
```

فعال کردن TLS. اگر mTLS لازم باشد، certfile/keyfile باید کلاینت باشند. برای self-signed میتوان گذاشت، ولی امنیت را کم میکند—بهترش دادن CA درست است.

## # Callbacks

```
c.on_connect = on_connect

c.on_disconnect = on_disconnect

c.enable_logger()
```

وصل کردن کالبکها و امکان لاغ داخلی.

## # Auto-reconnect backoff

```
c.reconnect_delay_set(min_delay=1, max_delay=60)
```

تنظیم بکآف برای تلاش‌های اتصال دوباره.

```
اتصال غیرمسدودکننده + شروع لوپ شبکه در ترد جدا

c.connect_async(MQTT["host"], MQTT["port"], keepalive=MQTT["keepalive"])

c.loop_start()
```

```
return c
```

اتصال `async` و شروع حلقه‌ی شبکه در ترد جدا تا برنامه‌ی اصلی بلاک نشود.

```
def main():
```

```
 global _connected
```

```
 client = make_client()
```

حلقه‌ی اصلی ارسال متريکها

ساخت کلاینت و شروع لوب.

```
صبر اوليه تا وصل شود (غيربلوككتنده)
```

```
boot_wait_s = 0
```

```
while not _connected and boot_wait_s < 10:
```

```
 time.sleep(0.5); boot_wait_s += 0.5
```

تا ۱۰ ثانие فرصت برای اتصال اوليه.

```
try:
```

```
 while True:
```

```
اگر قطع است، فقط صبر کن؛ loop_start خودش reconnect می‌کند
```

```
 if not _connected:
```

```
 time.sleep(1.0)
```

```
 continue
```

در زمان قطع شدن، بی‌کار و منتظر بمان (بدون کرش).

```
cpu = psutil.cpu_percent(interval=1.0) # 1s block
```

```

ram = psutil.virtual_memory().percent
temp = get_temp_c()
ip = get_ip()
rss = get_wifi_rssi()

```

نمونه برداری: `cpu_percent(interval=1)` یک ثانیه بلاک می‌کند (میانگین ۱ ثانیه‌ای).

```

payload} =
{
 "ts": now_iso_utc(), # ISO-8601 UTC Z
 "cpu_percent": round(cpu, 1),
 "ram_percent": round(ram, 1),
 "cpu_temp_c": round(temp, 1) if temp is not None else None,
 "ip_addr": ip,
 "wifi_rssi": rss,
 "host": HOSTNAME
}

```

ساخت JSON خروجی با زمان، CPU/RAM/Temp/IP/RSSI و نام میزبان.

```

info = client.publish(MQTT["topic_metrics"], json.dumps(payload),
qos=MQTT["qos"], retain=False)

if MQTT["qos"] == 1:
 info.wait_for_publish(timeout=2.0)

```

ارسال روی تاپیک متریک‌ها با `QoS=1` دریافت می‌شود). `wait_for_publish` تا ۲ ثانیه صبر می‌کند (اختیاری اما خوب).

# فاصله‌ی نمونه‌برداری

```
time.sleep(2.0) # مجموعاً ۳~۲ ثانیه با cpu_percent(interval=1)
```

فاصله‌ی نمونه‌برداری: تقریباً هر ۳~۲ ثانیه (۱ ثانیه از sleep + ۲cpu\_percent).

```
except KeyboardInterrupt:
```

```
 pass
```

```
finally:
```

```
 try:
```

# اعلام آفلاین و توقف تمیز

```
 client.publish(MQTT["topic_lwt"], json.dumps({"status": "offline", "ts": now_iso_utc()}),
 qos=1, retain=True)
```

```
except Exception:
```

```
 pass
```

```
client.loop_stop() # شبکه را متوقف کن()
```

```
client.disconnect()
```

خروج تمیز: LWT را "offline" می‌گذارد (retain)، لوب شبکه را می‌بندد و قطع اتصال.

```
if __name__ == "__main__":
```

```
 main()
```

اجرای مستقیم اسکریپت.

خلاصه‌ی عملکرد

یک کلاینت MQTT با TLS می‌سازد، LWT را تنظیم می‌کند، و به صورت غیربلوک‌کننده وصل می‌شود.

هر ۳ ثانیه متریک‌های CPU/RAM/Temp/IP/WiFi RSSI را جمع می‌کند و با QoS=1 روی تاپیک iot/pi- منتشر می‌کند.

وضعیت آنلاین/آفلاین را روی تاپیک LWT به صورت retain نگه می‌دارد.

قطع اتصال را هندل می‌کند و خودکار reconnect می‌شود.

# با تشکر