طراحی سامانه مدیریت هوشمند ساختمان وب‌بنیان با معماری سرور مستقل: توسعه پروتکل امن و بومی جهت حذف وابستگی ابری و بیشینه‌سازی کنترل

علی لطفی اربط[۱]
۱- گروه مهندسی کامپیوتر ، دانشگاه مهارت ملی تبریز
alilotfyarbat@gmail.com

چکیده

با گسترش روزافزون اینترنت اشیاء [۱] در صنعت ساختمان، وابستگی به پلتفرم‌های ابری خارجی [۲] به یکی از چالش‌های اصلی در حوزه‌های امنیت داده، تداوم سرویس و تأخیر شبکه تبدیل شده است. وابستگی به سرورهای واسط، ریسک‌های امنیتی و هزینه‌های نگهداری را افزایش می‌دهد. هدف این پژوهش، طراحی و پیاده‌سازی یک سامانه مدیریت هوشمند ساختمان [۳] (BMS) با معماری سرور مستقل و بومی است که بدون نیاز به زیرساخت‌های ابری و واسط‌های مخابراتی پیچیده، کنترل کامل تجهیزات و فرآیندها را فراهم آورد. در این راستا، یک معماری توزیع‌شده‌ی سه‌لایه با بهره‌گیری از پشته نرم‌افزاری لمپ [۴] طراحی گردید. نوآوری اصلی، توسعه یک پروتکل ارتباطی سبک و ناهمگام [۵] بر بستر استاندارد وب است که جایگزین پروتکل‌های سربار بالا شده است. نودهای سخت‌افزاری با استفاده از مکانیزم «سرکشی دوره‌ای بهینه شده» [۶] و الگوریتم «ضربان‌قلب نرم‌افزاری» [۷]، همگام‌سازی بلادرنگ داده‌ها را انجام می‌دهند. نتایج ارزیابی عملکرد سامانه با اجرای آزمون‌های بار بر روی ۲۵۰ کاربر همزمان، میانگین زمان پاسخ‌دهی ۳۴۰ میلی‌ثانیه و نرخ خطای صفر را نشان داد. همچنین، پیاده‌سازی مکانیزم‌های امنیتی دفاع در عمق شامل احراز هویت دوعاملی [۸] و کنترل دسترسی نقش‌محور [۹]، پایداری سیستم را تضمین نموده است.

کلمات کلیدی : سامانه مدیریت ساختمان، معماری سرور مستقل، پروتکل ارتباطی وب‌بنیان، اینترنت اشیاء، امنیت سایبری.

[۱]: Internet of Things (IoT)
[۲]: Third-party Cloud Platforms
[۳]: Building Management System
[٤]: LAMP Stack (Linux, Apache, MySQL, PHP)
[٥]: Asynchronous
[٦]: Optimized Periodic Polling
[۷]: Software Heartbeat
[۸]: Two-Factor Authentication / One-Time Password (OTP)
[۹]: Role-Based Access Control (RBAC)

۱ – مقدمه

در دهه‌های اخیر، رشد شتابان شهرنشینی و تغییر الگوهای سکونت به سمت مجتمع‌های مسکونی متراکم، مدیریت ساختمان‌ها را با چالش‌های نوظهوری مواجه ساخته است. در این بافت جدید، روش‌های سنتی مدیریت که بر فرآیندهای دستی و غیرمتمرکز استوارند، دیگر قادر به پاسخگویی به نیازهای نظارتی، امنیتی و عملیاتی نیستند. ظهور فناوری اینترنت اشیاء و ادغام آن با سامانه‌های مدیریت ساختمان، پارادایم جدیدی را ایجاد کرده است که امکان گذار از مدیریت واکنشی [۱۰] به مدیریت پیشگیرانه [۱۱] و هوشمند را فراهم می‌آورد.

با وجود مزایای بالقوه، پیاده‌سازی این سامانه‌ها در زیرساخت‌های فعلی با موانع فنی و امنیتی متعددی روبروست. یکی از چالش‌های بنیادین در راهکارهای موجود، وابستگی شدید به پلتفرم‌های ابری خارجی و پروتکل‌های ارتباطی پیچیده نظیر MQTT [۱۲] است .این وابستگی نه تنها مخاطراتی را در زمینه حریم خصوصی و امنیت داده‌ها ایجاد می‌کند، بلکه در شرایط ناپایداری شبکه یا اختلال در اینترنت جهانی، تداوم سرویس‌دهی را مختل می‌سازد .علاوه بر این، بسیاری از سامانه‌های داخلی موجود فاقد رویکرد یکپارچه بوده و اغلب به صورت جزیره‌ای عمل می‌کنند؛ به طوری که یا صرفاً بر جنبه‌های مالی تمرکز دارند و یا فاقد زیرساخت لازم برای کنترل بلادرنگ تجهیزات سخت‌افزاری هستند.

این پژوهش با درک این چالش‌ها، طراحی و پیاده‌سازی یک سامانه مدیریت هوشمند ساختمان کاملاً بومی و وب‌بنیان را ارائه می‌دهد که بر پایه معماری «سرور مستقل» بنا شده است . رویکرد اصلی در این معماری، حذف واسط‌های ابری و انتقال قدرت پردازشی و ذخیره‌سازی داده‌ها به سرورهای محلی است. این راهکار، علاوه بر تضمین مالکیت داده‌ها و افزایش امنیت، تأخیر شبکه [۱۳] را کاهش داده و پایداری سیستم را در شبکه‌های داخلی (اینترانت) تضمین می‌کند.

نوآوری فنی این مقاله در توسعه یک پروتکل ارتباطی ناهمگام و سبک بر بستر پروتکل استاندارد انتقال ابرمتن (HTTP) است .برخلاف روش‌های مرسوم که نیازمند راه‌اندازی کارگزار [۱۴] برای تبادل پیام هستند، در این طرح از مکانیزم «سرکشی دوره‌ای بهینه‌شده» و الگوریتم تشخیص وضعیت «ضربان‌قلب نرم‌افزاری» استفاده شده است. این پروتکل که بر روی ریزکنترلگرهای ESP۳۲ و بستر نرم‌افزاری سرور پیاده‌سازی شده، امکان پایش وضعیت اتصال و کنترل تجهیزاتی نظیر روشنایی و سنسورهای پارکینگ را با پایداری بالا و بدون سربار اضافی فراهم می‌آورد[۱].

در ادامه این مقاله، در بخش دوم به تشریح مواد و روش‌ها شامل معماری سخت‌افزار، پشته نرم‌افزاری و پروتکل پیشنهادی پرداخته می‌شود . بخش سوم به جزئیات پیاده‌سازی فنی، چالش‌های کدنویسی و ساختار الگوریتم‌ها اختصاص دارد. در بخش چهارم، نتایج حاصل از آزمون‌های عملکردی، تست بار و ارزیابی‌های امنیتی ارائه می‌گردد. در نهایت، در بخش پنجم نتیجه‌گیری کلی پژوهش بیان شده و در بخش ششم، پیشنهاداتی برای مطالعات آتی ارائه خواهد شد.

[۱۰]: Reactive Management
[۱۱]: Proactive Management
[۱۲]: Message Queuing Telemetry Transport
[۱۳]: Network Latency
[۱۴]: Broker

۲- مواد و روش‌ها

این پژوهش با رویکردی توسعه‌ای-کاربردی و بر مبنای معماری نرم‌افزاری سه‌لایه (شامل لایه فیزیکی، لایه منطق و لایه نمایش) انجام شده است .سیستم پیشنهادی جهت دستیابی به اهداف استقلال و امنیت، از ترکیب سخت‌افزارهای تعبیه‌شده و سرویس‌های وب محلی بهره می‌برد[۱].

۲-۱ . معماری سخت‌افزار و تجهیزات

در لایه فیزیکی، از ریزکنترلر [۱۵] مدل ESP۳۲ به عنوان نود پردازشی مرکزی استفاده شده است .انتخاب این برد به دلیل برخورداری از پشته شبکه وای‌فای داخلی و پردازنده قدرتمند دو هسته‌ای جهت مدیریت همزمان حسگرها و ارتباطات شبکه صورت گرفت. تجهیزات متصل به این نود عبارتند از:

- حسگرهای فراصوت :[۱۶] مدل HC-SR۰۵ جهت تشخیص وضعیت اشغال پارکینگ با اندازه‌گیری زمان رفت و برگشت پالس صوتی، متصل به پایه‌های دیجیتال ورودی.
- ماژول‌های رله :[۱۷] جهت کنترل زون‌های روشنایی ساختمان که به پایه‌های خروجی دیجیتال متصل شده و فرمان‌های سرور را اجرا می‌کنند.
- نشانگرهای وضعیت محلی :جهت اعلام هشدار در محل (مانند پر بودن پارکینگ) مستقل از اتصال شبکه.

۲-۲ .پشته نرم‌افزاری سمت سرور

بخش منطق و داده‌ها با هدف حذف وابستگی به سرویس‌های ابری شخص ثالث، بر روی یک سرور لینوکس محلی با پشته نرم‌افزاری لمپ پیاده‌سازی گردید.

مدیریت داده‌ها :جهت ذخیره‌سازی و بازیابی اطلاعات، از پایگاه داده MySQL با ساختار رابطه‌ای نرمال‌سازی شده[۱۸] تا سطح سوم (۳ N) استفاده شد .تمامی تراکنش‌های حساس (مانند ثبت وضعیت‌ها و امور مالی) با رعایت اصول ACID [۱۹] مدیریت می‌شوند.

- منطق برنامه :رابط‌های برنامه‌نویسی (API) با استفاده از زبان PHP توسعه یافتند .جهت ایمن‌سازی لایه داده، تمامی کوئری‌ها با استفاده از «دستورات آماده» در کتابخانه PDO اجرا می‌شوند تا از نفوذ کد مخرب جلوگیری شود.
- رابط کاربری :لایه نمایش با رویکرد وب‌اپلیکیشن پیش‌رونده (PWA) [۲۰] طراحی شد تا تجربه‌ای مشابه نرم‌افزارهای بومی را بر روی مرورگر ارائه دهد .در توسعه این بخش از فریم‌ورک‌های مدرن برای طراحی واکنش‌گرا و مدیریت تعاملات ناهمگام استفاده شده است.

---

[۱۵]: Micro Controller
[۱۶]: Ultrasonic Sensors
[۱۷]: Relays
[۱۸]: Normalization
[۱۹]: Atomicity, Consistency, Isolation, Durability (ACID)
[۲۰]: Progressive Web App (PWA)

۲-۳ .پروتکل ارتباطی بومی

به جای استفاده از پروتکل‌های استاندارد اما نیازمند زیرساخت واسط مانند(MQTT)، یک پروتکل ارتباطی سفارشی بر بستر HTTPS طراحی شد .در این پروتکل، نود سخت‌افزاری در دو حلقه زمانی مجزا فعالیت می‌کند:

۱. حلقه بلادرنگ :با فرکانس ۱ هرتز، داده‌های محیطی (مانند فاصله خودرو) قرائت شده و منطق‌های محلی (مانند روشن شدن چراغ هشدار در فاصله کمتر از ۳ سانتی‌متر) اجرا می‌شوند.
۲. حلقه شبکه :با فرکانس ۰٫۲ هرتز (هر ۵ ثانیه)، وضعیت سنسورها به سرور ارسال شده و وضعیت جدید عملگرها دریافت می‌گردد.

مکانیزم پایش وضعیت : برای حل چالش عدم اتصال دائم در پروتکل وب، الگوریتم «ضربان‌قلب نرم‌افزاری» در سمت سرور پیاده‌سازی شد. نود سخت‌افزاری در هر بار اتصال، برچسب زمانی آخرین فعالیت خود را در پایگاه داده به‌روزرسانی می‌کند. سرور با مقایسه زمان فعلی و آخرین زمان ثبت‌شده، وضعیت آنلاین یا آفلاین بودن دستگاه را با آستانه ۲۰ ثانیه تشخیص می‌دهد.

۲-۴ .روش‌های ارزیابی و آزمون

جهت سنجش کارایی و قابلیت اطمینان سامانه، از استراتژی آزمون چندسطحی استفاده شد:

- آزمون عملکردی :بیش از ۸۰ سناریوی آزمون شامل بررسی صحت عملکرد حسگرها، تغییر وضعیت رله‌ها و تراکنش‌های همزمان اجرا گردید.
- آزمون بار :[۲۱] پایداری سرور و پروتکل ارتباطی با شبیه‌سازی ۲۵۰ کاربر همزمان[۲۲] مورد ارزیابی قرار گرفت تا عملکرد سیستم در شرایط ترافیک بالا سنجیده شود.
- آزمون نفوذ :[۲۳] امنیت سامانه در برابر آسیب‌پذیری‌های شناخته شده وب (OWASP Top ۱۰) از جمله تزریق SQL و شکست احراز هویت مورد ممیزی قرار گرفت[۱].

۳- پیاده‌سازی و جزئیات فنی

در این بخش، فرآیند تبدیل معماری انتزاعی به سامانه‌ای عملیاتی تشریح می‌گردد. پیاده‌سازی این پژوهش شامل توسعه سفت‌افزار برای نودهای سخت‌افزاری، برنامه‌نویسی سمت سرور و طراحی الگوریتم‌های مدیریت وضعیت است.

۳-۱ . محیط توسعه و ابزارها

برای توسعه و استقرار سامانه، از پشته نرم‌افزاری لمپ (لینوکس، آپاچی، مای‌اس‌کی‌وال، پی‌اچ‌پی) استفاده شد که بستری پایدار و متن‌باز را فراهم می‌آورد . کدنویسی سمت سرور با استفاده از ویرایشگر VS Code و مدیریت وابستگی‌ها توسط ابزار کامپوزر انجام گرفت . برای برنامه‌ریزی ریزکنترلر ESP۳۲ نیز از محیط توسعه آردوینو بهره گرفته شد که امکان استفاده از کتابخانه‌های بهینه‌شده C++ را فراهم می‌کند.

[۲۱]: Load Testing
[۲۲]: Concurrent Users
[۲۳]: Penetration Testing

۳-۲ .پیاده‌سازی منطق سخت‌افزار

هسته پردازشی در لایه فیزیکی، کدهای توسعه‌یافته برای ESP۳۲ است. برخلاف روش‌های معمول که از وقفه‌های مسدودکننده استفاده می‌کنند، در این پژوهش یک منطق «غیرمسدودکننده «بر پایه زمان‌سنج داخلی سیستم طراحی شد. ساختار کد شامل دو حلقه کنترلی موازی است:

- **حلقه سریع (مدیریت حسگرها) :**این حلقه با تناوب ۱ ثانیه اجرا می‌شود. در هر تکرار، تابع CheckParkingSensor با ارسال پالس به حسگر فراصوت، فاصله را محاسبه می‌کند. اگر فاصله از آستانه تعیین‌شده (۳ سانتی‌متر) کمتر باشد، وضعیت «اشغال» تشخیص داده شده و چراغ هشدار محلی فعال می‌گردد.

- **حلقه شبکه (همگام‌سازی) :**این حلقه با تناوب ۵ ثانیه اجرا می‌شود تا سربار شبکه کاهش یابد. تابع CheckLightStatus با ارسال درخواست به سرور، آرایه جی‌سون [۲۴] حاوی وضعیت زون‌های روشنایی را دریافت، پردازش و بر روی پین‌های خروجی اعمال می‌کند.

۳-۳ .پیاده‌سازی رابط‌های سمت سرور

منطق تجاری سامانه با استفاده از زبان PHP و به صورت شی‌گرا پیاده‌سازی شد. فایل‌های کنترلی به عنوان نقاط پایانی رابط برنامه‌نویسی[۲۵] عمل می‌کنند. برای نمونه، ماژول مدیریت پارکینگ، درخواست‌های دریافتی از سخت‌افزار را اعتبارسنجی کرده و داده‌ها (شامل شناسه جایگاه و فاصله) را در پایگاه داده ثبت می‌کند . جهت حفظ یکپارچگی داده‌ها در محیط‌های چندکاربره، تمامی عملیات حساس (مانند ثبت تراکنش‌های مالی و تغییر وضعیت تجهیزات) درون تراکنش‌های پایگاه داده[۲۶] اجرا می‌شوند تا از بروز ناهماهنگی در صورت قطع ناگهانی عملیات جلوگیری شود.

۳-۴ .چالش‌های فنی و الگوریتم‌های کلیدی

در طول فرآیند پیاده‌سازی، چالش‌های متعددی شناسایی و مرتفع گردید:

الف) چالش تأخیر و همگام‌سازی :یکی از معایب پروتکل‌های مبتنی بر وب، تأخیر[۲۷] ذاتی در روش سرکشی (Polling) است. برای حل این چالش، الگوریتم سمت سرور به گونه‌ای بهینه‌سازی شد که تنها در صورت تغییر وضعیت (تغییر داده)، پاسخ کامل ارسال شود. همچنین فرکانس ارسال درخواست‌ها بر اساس پهنای باند شبکه داخلی تنظیم گردید تا تعادلی میان سرعت پاسخ‌دهی و بار سرور ایجاد شود.

ب) الگوریتم ضربان‌قلب نرم‌افزاری :از آنجا که پروتکل وب فاقد اتصال دائم است، تشخیص قطعی سخت‌افزار با چالش مواجه بود. برای رفع این مشکل، الگوریتم «ضربان‌قلب» پیاده‌سازی شد. در این روش، نود سخت‌افزاری در هر بار ارتباط موفق، برچسب زمانی last_updated را در پایگاه داده به‌روزرسانی می‌کند. اسکریپت ناظر در سمت سرور، با مقایسه زمان جاری و آخرین زمان ثبت‌شده، وضعیت اتصال را پایش کرده و در صورت عبور اختلاف زمانی از ۲۰ ثانیه، وضعیت دستگاه را به «آفلاین» تغییر می‌دهد[۱].

---

[۲٤]: JavaScript Object Notation
[۲۰]: API Endpoints
[۲٦]: Database Transactions
[۲۷]: Latency

بر اساس تحلیل ارائه‌شده در جدول فوق، برتری راهبردی پروتکل وب‌بنیان پیشنهادی نسبت به استاندارد MQTT در کاربردهای خاص مدیریت ساختمان، در **استقلال عملیاتی** و **سادگی ذاتی** آن نهفته است.

در حالی که استفاده از MQTT اغلب مستلزم پذیرش پیچیدگی راه‌اندازی کارگزارها و ریسک وابستگی به پلتفرم‌های ابری خارجی است ، پروتکل پیشنهادی با حذف کامل نیاز به واسطهای مخابراتی پیچیده، قدرت کنترل و پردازش را تماماً به زیرساخت محلی منتقل می‌کند .این رویکرد، نه تنها امنیت داده‌ها و حریم خصوصی را با تضمین مالکیت اطلاعات ارتقا می‌دهد ، بلکه پایداری سیستم و تداوم سرویس‌دهی را حتی در شرایط ناپایداری شبکه یا قطع اینترنت جهانی تضمین می‌نماید.

۴ - نتایج و یافته ها

جهت اطمینان از صحت عملکرد منطق سامانه، بیش از ۸۰ سناریوی آزمون بر روی ماژول‌های مختلف اجرا گردید. جدول ۱ خلاصه نتایج آزمون‌های عملکردی کلیدی را نشان می‌دهد. همانطور که مشاهده می‌شود، جداسازی لایه منطق از لایه نمایش، منجر به پایداری عملکرد در تمامی سناریوها شده است.

جدول ۱: خلاصه نتایج آزمون‌های عملکردی ماژول‌های کلیدی

شناسه آزمون	ماژول تحت آزمون	شرح سناریوی آزمون	نتیجه مشاهده شده	وضعیت
TC-AU-۰۱	احراز هویت	ورود کاربر جدید به سامانه پس از تأیید موفق کد یکبار مصرف (OTP)	هدایت خودکار کاربر به صفحه تکمیل ثبت‌نام و ایجاد نشست کاربری	موفق
TC-AC-۰۲	کنترل دسترسی	تلاش کاربر با نقش «ساکن» برای دسترسی مستقیم به URL پنل «مدیر»	مسدودسازی درخواست توسط سیستم و هدایت کاربر به صفحه ورود	موفق
TC-IO-۰۳	اینترنت اشیاء	ارسال فرمان تغییر وضعیت روشنایی از طریق پنل وب به سخت‌افزار	تغییر وضعیت در دیتابیس و اعمال فرمان روی سخت‌افزار با تأخیر کمتر از ۱ ثانیه	موفق
TC-SE-۰۴	امنیت (XSS)	تزریق اسکریپت مخرب <script> در فیلدهای متنی ورودی	نمایش ورودی به صورت متن ساده و خنثی‌سازی شده (Sanitized)	موفق

یکی از چالش‌های اصلی پروتکل‌های مبتنی بر وب، سربار احتمالی بر روی سرور در ترافیک بالاست. برای ارزیابی پایداری معماری پیشنهادی، آزمون بار با استفاده از ابزار Loader.io اجرا گردید. در این آزمون، تعداد ۲۵۰ کاربر همزمان به مدت ۵۰ ثانیه به سرور درخواست ارسال کردند.

نتایج حاصله نشان می‌دهد که میانگین زمان پاسخ‌دهی ۳۴۰ میلی‌ثانیه است . این عدد اثبات می‌کند که پروتکل اختصاصی طراحی شده، با وجود استفاده از روش سرکشی دوره‌ای، سربار پردازشی قابل قبولی دارد و نرخ خطای ٪۰,۰۰ نشان‌دهنده پایداری کامل سرور است.

۴-۳ .ارزیابی کیفیت تجربه کاربری[۱]

کیفیت رابط کاربری وب‌بنیان با استفاده از ابزار استاندارد Google Lighthouse مورد ممیزی قرار گرفت. کسب امتیاز ۹۲ در شاخص عملکرد[۲۸] و ۱۰۰ در شاخص سئو[۲۹] نشان‌دهنده بهینه‌سازی کدهای سمت کاربر و استفاده صحیح از کش‌گذاری است . همچنین، در ارزیابی میدانی با استفاده از پرسشنامه استاندارد مقیاس قابلیت استفاده(SUS) ، سامانه موفق به کسب امتیاز ۹۴ از ۱۰۰ گردید که در رده «عالی» طبقه‌بندی می‌شود.

۴-۴ .تحلیل امنیت و پایداری

امنیت سامانه در برابر ۱۰ آسیب‌پذیری برتر وب (OWASP Top ۱۰) مورد ارزیابی قرار گرفت. نتایج آزمون نفوذپذیری در جدول ۲ خلاصه شده است. استفاده از دستورات آماده در لایه دسترسی به داده، به طور کامل از حملات تزریق SQL جلوگیری کرده است.

جدول ۲: نتایج ارزیابی امنیتی در برابر تهدیدات سایبری

ردیف	عنوان آسیب‌پذیری	راهکار پیاده‌سازی شده و نتایج ارزیابی سامانه	وضعیت
۰۱A	نقص در کنترل دسترسی	ارزیابی‌ها نشان می‌دهد که سامانه با بررسی سطح دسترسی قبل از بارگذاری داشبوردها و محدودسازی رابط‌های برنامه‌نویسی (API) به درخواست‌های معتبر، از دسترسی‌های غیرمجاز جلوگیری می‌کند.	✓ ایمن
۰۲A	شکست‌های رمزنگاری	آزمون شنود شبکه تایید می‌کند که با به‌کارگیری پروتکل انتقال ابرمتن امن (HTTPS) و ماژول‌های رمزنگاری در سخت‌افزار، کانال تبادل داده در برابر شنود محافظت شده است.	✓ ایمن
۰۳A	تزریق کد (داده و اسکریپت)	نتایج تست نفوذ نشان‌دهنده مقاومت سامانه در برابر حملات تزریق است؛ این امر از طریق مکانیزم «دستورات آماده» برای پایگاه داده و پاک‌سازی ورودی‌ها جهت خنثی‌سازی اسکریپت‌های مخرب محقق شده است.	✓ ایمن
۰۴A	طراحی ناامن	معماری سامانه بر اساس اصول «امنیت در طراحی» بنا شده و با جداسازی لایه منطق تجاری از لایه رابط کاربری، سطح حمله به حداقل رسیده است.	✓ ایمن

[۲۸]: Performance Index
[۲۹]: SEO (Search Engine Optimization)

ردیف	عنوان آسیب‌پذیری	راهکار پیاده‌سازی شده و نتایج ارزیابی سامانه	وضعیت
۰۵A	پیکربندی امنیتی نادرست	بررسی پاسخ‌های سرور نشان می‌دهد که مدیریت خطاها به صورت داخلی انجام شده و از نشت اطلاعات حساس پیکربندی یا ساختار پایگاه داده به کاربر نهایی جلوگیری می‌شود.	✓ ایمن
۰۶A	اجزای آسیب‌پذیر و قدیمی	پایش وابستگی‌های نرم‌افزاری حاکی از آن است که سامانه با استفاده از ابزارهای مدیریت بسته استاندارد، همواره از کتابخانه‌های به‌روز و فاقد آسیب‌پذیری‌های شناخته‌شده استفاده می‌کند.	✓ ایمن
۰۷A	شکست در شناسایی و احراز هویت	با جایگزینی رمزهای عبور ثابت با سیستم احراز هویت مبتنی بر رمز یکبار مصرف (OTP) و مدیریت صحیح نشست‌های کاربری، ریسک جعل هویت مرتفع گردیده است.	✓ ایمن
۰۸A	نقص در یکپارچگی داده و نرم‌افزار	آزمون‌های صحت‌سنجی داده‌ها تایید می‌کند که استفاده از تراکنش‌های اتمیک (ACID)، یکپارچگی اطلاعات ثبت‌شده از سنسورها را حتی در شرایط ترافیک بالا تضمین نموده است.	✓ ایمن
۰۹A	نارسایی در لاگ‌گیری و پایش	سامانه با بهره‌گیری از الگوریتم‌های پایش وضعیت (Heartbeat) و ثبت زمان دقیق فعالیت نودها، امکان تشخیص بلادرنگ تجهیزات معیوب یا قطع‌شده را فراهم کرده است.	✓ ایمن
۱۰A	جعل درخواست سمت سرور	اعتبارسنجی سخت‌گیرانه پارامترهای ورودی در لایه کنترلر، مانع از ارسال درخواست‌های ساختگی یا دسترسی به منابع داخلی شبکه توسط مهاجمین شده است.	✓ ایمن

ارزیابی امنیتی سامانه نه صرفاً با بازبینی کد، بلکه از طریق اجرای سناریوهای حمله شبیه‌سازی شده انجام پذیرفته است. همان‌طور که در جدول ۲ مشاهده می‌شود، با ادغام راهکارهای دفاع در عمق، تمامی ۱۰ آسیب‌پذیری بحرانی معرفی شده توسط بنیاد OWASP در نسخه ۲۰۲۱، در لایه‌های مختلف سخت‌افزار و نرم‌افزار پوشش داده شده‌اند.

۵-۴ .تحلیل مقایسه‌ای پروتکل پیشنهادی با استانداردMQTT

در این بخش، پروتکل ارتباطی وب‌بنیان توسعه‌یافته در این پژوهش با پروتکل استاندارد و پرکاربرد MQTT در جدول (۳) مورد مقایسه دقیق قرار می‌گیرد. همان‌طور که در جدول ویژگی‌ها خلاصه شده است، این دو پروتکل در معماری پایه، نیازمندی‌های زیرساختی و رویکرد مدیریت داده تفاوت‌های بنیادینی دارند که هر یک را برای سناریوهای خاصی مناسب می‌سازد.

تفاوت‌های معماری و زیرساختی :بر اساس معیارهای ۱ و ۲ جدول ، اساسی‌ترین تفاوت در مدل ارتباطی و نیاز به واسط است. پروتکل پیشنهادی از مدل «درخواست-پاسخ (سرکشی)» استفاده می‌کند که در آن سخت‌افزار کلاینت باید فعالانه برای دریافت اطلاعات جدید به سرور مراجعه کند. این رویکرد منجر به معماری بدون واسط شده است که در آن سخت‌افزار مستقیماً با وب‌سرور استاندارد مانند آپاچی تعامل دارد و نیاز به نصب و نگهداری نرم‌افزارهای جانبیِ کارگزارمانند

Mosquitto که در MQTT الزامی است، حذف می‌شود. در مقابل،MQTT بر پایه مدل «انتشار-اشتراک» بنا شده که نیازمند یک کارگزار مرکزی برای توزیع پیام‌ها است.

مدیریت اتصال و سازگاری با شبکه :معیارهای ۳ و ۴ نشان می‌دهند که پروتکل پیشنهادی ناهمگام و بدون حالت است و اتصال دائمی برقرار نمی‌کند؛ در عوض، وضعیت آنلاین بودن دستگاه از طریق مکانیزم «ضربان‌قلب نرم‌افزاری» و ثبت زمان در پایگاه داده کنترل می‌شود. این در حالی است که MQTT یک اتصال TCP دائم و متصل را حفظ می‌کند. مزیت کلیدی رویکرد پیشنهادی، سازگاری بسیار عالی با فایروال‌ها به دلیل استفاده از پورت‌های استاندارد وب (۴۴۳/۸۰) است، در حالی که MQTT ممکن است در شبکه‌های دارای محدودیت امنیتی به دلیل نیاز به پورت‌های خاص (مانند ۱۸۸۳) با چالش مواجه شود.

کارایی، بلادرنگی و مصرف انرژی :در بررسی معیارهای ۵، ۸ و ۹، تقابل بین سادگی و کارایی بلادرنگ آشکار می‌شود. پروتکل MQTT به دلیل داشتن سرایندهای باینری بسیار سبک، سربار پردازشی بسیار کمی دارد و به دلیل اتصال دائم، قابلیت بلادرنگ واقعی را ارائه می‌دهد که در شبکه‌های پایدار مصرف انرژی پایین‌تری نیز دارد. در مقابل، پروتکل پیشنهادی دارای سربار متوسط به دلیل هدرهای HTTP و ماهیت «شبه‌بلادرنگ» است، زیرا سرعت دریافت فرمان‌ها وابسته به فاصله زمانی سرکشی (مثلاً ۵ ثانیه) است. همچنین، به دلیل نیاز به برقراری مجدد اتصال در هر بار سرکشی، مصرف انرژی در کلاینت نسبتاً بالاتر است.

توسعه‌پذیری و یکپارچگی داده :طبق معیارهای ۶ و ۷، پروتکل پیشنهادی از نظر پیاده‌سازی و نگهداری «کم و ساده» است زیرا بر دانش استاندارد توسعه وب متکی است و با اتصال مستقیم به پایگاه داده رابطه‌ای، تضمین یکپارچگی داده را برای تراکنش‌های حساس فراهم می‌کند MQTT .در مقابل، نیازمند دانش تخصصی درباره پروتکل و مدیریت کارگزار است (پیچیدگی متوسط تا زیاد) و اگرچه با سطوح کیفیت سرویس (QOS) اطمینان از تحویل پیام را فراهم می‌کند، اما ذاتاً داده‌محور و متصل به پایگاه داده نیست.

در نهایت، معیار ۱۰ نشان می‌دهد که هر دو پروتکل از TLS/SSL برای امنیت لایه انتقال پشتیبانی می‌کنند، اما پروتکل پیشنهادی بر مکانیزم‌های استاندارد احراز هویت وب متکی است، در حالی که MQTT از مکانیزم‌های اختصاصی خود در سمت کارگزار بهره می‌برد.

جدول (۳) مقایسه جامع: پروتکل پیشنهادی وب‌بنیان در برابر استانداردMQTT

ویژگی / معیار	پروتکل پیشنهادی در مقاله (وب‌بنیان / Web-Based)	پروتکل استاندارد MQTT
۱. معماری ارتباطی	مدل درخواست-پاسخ (سرکشی): کلاینت (سخت‌افزار) باید فعالانه و به صورت دوره‌ای برای دریافت اطلاعات جدید از سرور سوال بپرسد[۲].	مدل انتشار-اشتراک (Pub/Sub): کلاینت‌ها فقط به یک موضوع گوش می‌دهند و کارگزار مرکزی پیام را به محض رسیدن توزیع می‌کند.

ویژگی / معیار	پروتکل پیشنهادی در مقاله (وب‌بنیان / Web-Based)	پروتکل استاندارد MQTT
۲. نیاز به زیرساخت واسط	بدون نیاز به واسط: سخت‌افزار مستقیماً با وب‌سرور (مانند آپاچی و زبان PHP) صحبت می‌کند. نیاز به نصب نرم‌افزار جانبی اضافی نیست[۳].	نیازمند کارگزار مرکزی: برای عملکرد شبکه، حتماً نیاز به نصب، راه‌اندازی و نگهداری سرویس‌های واسط مانند Mosquitto یا RabbitMQ است.
۳. مدیریت وضعیت اتصال	ناهمگام و بدون حالت: اتصال دائمی وجود ندارد. وضعیت آنلاین بودن دستگاه با مکانیزم «ضربان‌قلب نرم‌افزاری» و ثبت زمان در پایگاه داده چک می‌شود[۴].	دائم و متصل: یک اتصال TCP همیشه باز بین کلاینت و کارگزار برقرار است و زنده بودن اتصال توسط بسته‌های «زنده‌نگه‌دار» (Keep-Alive) پروتکل مدیریت می‌شود.
۴. سازگاری با فایروال و شبکه	بسیار عالی: از درگاه‌های (پورت‌های) استاندارد وب (۸۰ برای HTTP یا ۴۴۳ برای HTTPS) استفاده می‌کند که معمولاً در تمام شبکه‌های سازمانی و خانگی باز هستند[۵].	متوسط تا دشوار: نیاز به باز بودن درگاه‌های خاص (مانند ۱۸۸۳ یا ۸۸۸۳) دارد که ممکن است در برخی شبکه‌های دارای محدودیت امنیتی مسدود باشند و چالش ایجاد کنند.
۵. سربار داده و پردازش	متوسط: سرایندهای پروتکل HTTP نسبت به MQTT حجیم‌تر هستند، اما این سربار با تنظیم فاصله زمانی سرکشی (مثلاً هر ۵ ثانیه یکبار) مدیریت شده است[۶].	بسیار کم: دارای سرایندهای باینری بسیار سبک و فشرده است (حتی در حد ۲ بایت) که برای شبکه‌های بسیار ضعیف یا با پهنای باند محدود ایده‌آل است.
۶. مدیریت تراکنش و داده	تضمین یکپارچگی داده (ACID): چون مستقیماً به پایگاه داده رابطه‌ای متصل است، صحت و یکپارچگی تراکنش‌های داده (مانند ثبت مالی یا تغییر وضعیت حساس) تضمین می‌شود[۷].	کیفیت سرویس (QOS): دارای سطوح مختلف برای اطمینان از تحویل پیام (سطح ۰، ۱، ۲) است، اما ذاتاً داده‌محور و متصل به پایگاه داده نیست.
۷. پیچیدگی پیاده‌سازی و نگهداری	کم و ساده: با دانش استاندارد توسعه وب (مانند PHP و پروتکل HTTP) به راحتی قابل توسعه، عیب‌یابی و نگهداری است.	متوسط تا زیاد: نیاز به دانش تخصصی درباره خود پروتکل، پیکربندی کارگزار مرکزی و مدیریت موضوعات (Topics) دارد.
۸. قابلیت بلادرنگ	شبه‌بلادرنگ: سرعت دریافت فرمان‌ها وابسته به فاصله زمانی سرکشی است. همیشه یک تاخیر ذاتی بین ارسال فرمان و دریافت آن توسط دستگاه وجود دارد (مثلاً تا ۵ ثانیه).	بلادرنگ واقعی: به دلیل اتصال دائم، به محض اینکه پیامی به کارگزار برسد، در همان لحظه برای دستگاه‌های مشترک ارسال می‌شود.
۹. مصرف انرژی در کلاینت	نسبتاً بالاتر: به دلیل نیاز به برقراری مجدد اتصال کامل در هر بار سرکشی (مخصوصاً در حالت HTTPS)، مصرف پردازشی و انرژی ماژول کمی بیشتر است.	پایین‌تر (در شبکه پایدار): در شبکه‌های پایدار، به دلیل حفظ یک اتصال باز و ارسال بسته‌های بسیار کوچک برای زنده نگه داشتن آن، از نظر مصرف انرژی بهینه‌تر است.

ویژگی / معیار	پروتکل پیشنهادی در مقاله (وب‌بنیان / Web-Based)	پروتکل استاندارد MQTT
۱۰. مدل امنیت	امنیت استاندارد وب: متکی بر پروتکل‌های امنیتی لایه انتقال وب (TLS/SSL بر روی HTTPS) و مکانیزم‌های احراز هویت در لایه نرم‌افزار (مانند توکن‌ها یا کوکی‌ها) است.	امنیت اختصاصی پروتکل: علاوه بر پشتیبانی از TLS/SSL، امنیت غالباً از طریق تنظیم لیست‌های کنترل دسترسی (ACL) و احراز هویت نام کاربری/رمز عبور در سمت کارگزار مرکزی مدیریت می‌شود.

نتیجه‌گیری مقایسه :تحلیل فوق نشان می‌دهد که پروتکل پیشنهادی با پذیرش تأخیر جزئی در دریافت فرمان‌ها (شبه‌بلادرنگ) و سربار کمی بیشتر، توانسته است پیچیدگی زیرساخت را به شدت کاهش دهد، سازگاری با شبکه‌های موجود را تضمین کند و یکپارچگی تراکنش‌های داده را با استفاده از استانداردهای وب فراهم آورد. این ویژگی‌ها آن را برای کاربردهایی که نیاز به کنترل دقیق بلادرنگ ندارند اما سادگی پیاده‌سازی و قابلیت اطمینان تراکنش‌ها در اولویت است، به گزینه‌ای مطلوب تبدیل می‌کند.

۵ - نتیجه‌گیری

این پژوهش با هدف طراحی و پیاده‌سازی یک سامانه مدیریت هوشمند ساختمان (BMS) با رویکرد استقلال از پلتفرم‌های ابری و تمرکز بر امنیت داده‌ها انجام شد. نتایج حاصل از پیاده‌سازی عملی و ارزیابی‌های فنی نشان می‌دهد که معماری پیشنهادی سرور مستقل مبتنی بر PHP و سخت‌افزار ESP۳۲ توانسته است به طور موثر جایگزین راهکارهای وابسته به اینترنت جهانی شود. دستاوردهای کلیدی این پژوهش عبارتند از:

۱ .کارایی پروتکل بومی :توسعه پروتکل ارتباطی مبتنی بر «سرکشی دوره‌ای» بر بستر پروتکل استاندارد وب، ثابت کرد که برای کاربردهای کنترلی ساختمان، نیازی به پیچیدگی و سربار پروتکل‌های سنگین نظیر MQTT نیست . میانگین زمان پاسخ‌دهی ۳۴۰ میلی‌ثانیه در آزمون بار، گواهی بر چابکی و کارآمدی این پروتکل در شبکه‌های داخلی است.

۲ .پایداری و تشخیص خطا :چالش عدم اتصال دائم در پروتکل وب با پیاده‌سازی الگوریتم «ضربان‌قلب نرم‌افزاری» مرتفع گردید. این مکانیزم با دقت ۲۰ ثانیه، قطع شدن ارتباط نودهای سخت‌افزاری را تشخیص داده و قابلیت اطمینان سیستم را به سطح استانداردهای صنعتی ارتقاء داده است.

۳ .امنیت دفاع در عمق :معماری امنیتی چندلایه شامل احراز هویت مبتنی بر رمز یکبار مصرف (OTP) و کنترل دسترسی نقش‌محور(RBAC) ، سامانه را در برابر تهدیدات رایج سایبری مقاوم ساخته است . نتایج آزمون‌های نفوذ نشان داد که جداسازی لایه داده و استفاده از دستورات آماده، ریسک تزریق کد را به طور کامل حذف نموده است.

۴ .بومی‌سازی و استقلال :برخلاف نمونه‌های خارجی که با چالش‌های ناسازگاری با زیرساخت‌های بانکی و شبکه داخلی مواجه هستند، این سامانه قابلیت اجرا بر روی شبکه اینترانت ملی را دارد که تضمین‌کننده تداوم سرویس و حفظ حریم خصوصی کاربران است.

۶ - پیشنهاد برای مطالعات آتی

با توجه به اینکه این پژوهش موفق به ارائه یک پلتفرم پایه، ماژولار و توسعه‌پذیر برای مدیریت هوشمند ساختمان شده است، مسیر برای تحقیقات تکمیلی و ارتقای سامانه جهت پوشش محدودیت‌های فعلی و پاسخگویی به نیازهای پیچیده‌تر هموار است. در همین راستا، محورهای زیر برای فازهای بعدی توسعه پیشنهاد می‌گردد:

۱ .توسعه اپلیکیشن‌های بومی : برای تعامل عمیق‌تر اگرچه نسخه وب‌اپلیکیشن پیش‌رونده پیاده‌سازی شده، دسترسی پذیری خوبی را بدون نیاز به نصب فراهم می‌کند، اما برای بهره‌برداری کامل از پتانسیل تلفن‌های هوشمند به عنوان کلید دیجیتال و رابط اصلی کاربر، توسعه نرم‌افزارهای بومی برای سیستم‌عامل‌های اندروید و iOS ضروری است. این امر امکانات زیر را فراهم می‌سازد:

- دسترسی به سخت‌افزار سطح پایین :استفاده از رابط های بومی برای دسترسی به بلوتوث کم‌مصرف جهت پیاده‌سازی سناریوهای ورود بدون لمس که با نزدیک شدن کاربر به درب، قفل به صورت خودکار باز شود.
- احراز هویت بیومتریک پیشرفته :ادغام مستقیم با ماژول‌های امنیتی سیستم‌عامل مانند تشخیص چهره یا اثر انگشت برای تأیید دو مرحله‌ای عملیات حساس یا ورود سریع و امن به اپلیکیشن.
- اعلان‌های آنی و حیاتی : ارسال هشدارهای بلادرنگ و قابل‌اتکا در مواقع اضطراری (مانند تشخیص دود یا نشت آب) که حتی در صورت بسته بودن برنامه در پس‌زمینه، کاربر را مطلع سازد؛ قابلیتی که در PWA محدودیت‌هایی دارد.

۲ .یکپارچه‌سازی با هوش مصنوعی و تحلیل داده‌محور: پلتفرم فعلی داده‌های ارزشمندی تولید می‌کند. گام بعدی، گذار از «کنترل دستی/برنامه‌ریزی‌شده» به «کنترل هوشمند و تطبیق‌پذیر» با افزودن لایه تحلیل داده و یادگیری ماشین است:

- مدیریت انرژی هوشمند :استفاده از الگوریتم‌های یادگیری تقویتی برای تحلیل الگوی مصرف ساکنین و شرایط محیطی، جهت تنظیم خودکار سیستم‌های سرمایش/گرمایش و روشنایی با هدف مینیمم‌سازی مصرف انرژی بدون کاهش سطح آسایش.
- نگهداری پیشگویانه : تحلیل داده‌های سری زمانی دریافتی از حسگرهای موتورخانه و تأسیسات (مانند لرزش، دما و جریان مصرفی) برای تشخیص ناهنجاری‌ها و پیش‌بینی خرابی تجهیزات قبل از وقوع، که منجر به کاهش چشمگیر هزینه‌های تعمیرات اضطراری و زمان توقف سیستم می‌شود.
- شخصی‌سازی تطبیقی :یادگیری عادات و ترجیحات هر کاربر در طول زمان برای پیشنهاد یا اعمال خودکار سناریوهای محیطی (مانند تنظیم شدت نور در ساعات خاصی از روز).

۳ .بازمهندسی معماری برای مقیاس‌پذیری کلان : معماری فعلی برای ساختمان‌های تک‌واحدی یا مجتمع‌های کوچک تا متوسط مناسب است. اما برای استقرار در شهرک‌های مسکونی یا برج‌های چند هزار واحدی که با حجم عظیمی از درخواست‌های همزمان و داده‌های حسگر مواجه هستند، تغییرات زیرساختی زیر پیشنهاد می‌شود:

- مهاجرت به معماری میکروسرویس‌ها : شکستن ساختار یکپارچه سرور فعلی به سرویس‌های مستقل و کوچک (مانند سرویس احراز هویت، سرویس ثبت داده، سرویس موتور قوانین، سرویس مدیریت دستگاه‌ها). این امر امکان مقیاس‌پذیری افقی اجزای پربار را با استفاده از فناوری‌های کانتینری مانند داکر فراهم می‌کند.

- استفاده از پروتکل‌های سبک‌تر در لایه لبه :در محیط‌هایی با هزاران گره حسگر، پروتکل‌های مبتنی بر TCP/HTTP سربار بالایی دارند. بررسی استفاده ترکیبی از پروتکل برنامه‌های محدودشده که مبتنی بر UDP و بسیار سبک است برای ارتباط حسگرهای کم‌توان با هاب مرکزی، و استفاده از MQTT یا HTTP فقط برای لایه‌های بالاتر، می‌تواند ترافیک شبکه را به شدت بهینه کند.
-

۴ .توسعه مفهوم دوقلوی دیجیتال و یکپارچگی : برای ارتقای سطح مدیریت بهره‌برداری و تعمیرات در ساختمان‌های بزرگ، ایجاد یک همزاد یا دوقلوی دیجیتال از ساختمان پیشنهاد می‌شود. این مدل مجازی که به‌صورت بلادرنگ با داده‌های حسگرهای سامانه BMS به‌روزرسانی می‌شود، امکانات زیر را فراهم می‌کند:

- شبیه‌سازی سناریوها :مدیران ساختمان می‌توانند پیش از اعمال تغییرات واقعی (مثلاً تغییر استراتژی تهویه مطبوع در تابستان)، تأثیر آن را بر مصرف انرژی و آسایش دمایی در مدل دیجیتال شبیه‌سازی و ارزیابی کنند.
- یکپارچگی با مدل اطلاعات ساختمان :اتصال سامانه هوشمند به نقشه‌های ساختمان، به طوری که در صورت بروز هشدار خرابی در یک تأسیسات، مکان دقیق آن روی نقشه سه‌بعدی به همراه تاریخچه تعمیرات و مشخصات فنی قطعه به تیم تأسیسات نمایش داده شود. این امر سرعت و دقت عملیات نگهداری را به طرز چشمگیری افزایش می‌دهد.

تشکر و قدردانی

اکنون که این پژوهش به سرانجام رسیده است، بر خود فرض می‌دانم مراتب سپاس و امتنان عمیق قلبی خویش را تقدیم اساتید گرانقدر و اعضای محترم هیئت علمی **گروه مهندسی کامپیوتر دانشگاه مهارت ملی تبریز** نمایم.

بی‌شک، به ثمر نشستن این تلاش علمی بدون بهره‌مندی از دانش وافر، بینش علمی ژرف و رهنمودهای داهیانه ایشان میسر نمی‌گردید. مشاوره‌های تخصصی و نقدهای سازنده این بزرگواران، چه در تبیین مبانی نظری و تدوین چارچوب پژوهش و چه در هدایت مسیر فنی جهت عبور از چالش‌های پیچیده پیاده‌سازی، چراغ راه من بوده است.

به‌ویژه، از حمایت‌های بی‌دریغ این گروه آموزشی و مساعدت مسئولین محترم در تخصیص منابع و فراهم‌سازی **زیرساخت‌های پردازشی و تجهیزات سخت‌افزاری پیشرفته** که بستر عملیاتی لازم را برای پیاده‌سازی کارآمد مدل‌های پیشنهادی و انجام ارزیابی‌های عملکردی دقیق سامانه در شرایط واقعی مهیا ساخت صمیمانه سپاسگزارم. استخراج نتایج قابل‌اتکا و اعتبارسنجی نهایی یافته‌های این پژوهش، مرهون این پشتیبانی‌های همه‌جانبه علمی و عملی است.

مراجع

١. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (٢٠١٥). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, ١٧(٤), ٢٣٤٧-٢٣٧٦.

٢. Espressif Systems. (٢٠٢٤). *ESP32 Series Technical Reference Manual (Version 5.1)*. Shanghai: Espressif Systems. Retrieved from .

٣. Fielding, R. T. (٢٠٠٠). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation). University of California, Irvine.

٤. Lea, P. (٢٠٢٠). *Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, and analytics*. Packt Publishing Ltd.

٥. OWASP Foundation. (٢٠٢١). *OWASP Top 10: 2021 – The Ten Most Critical Web Application Security Risks*. Retrieved from .

٦. Pressman, R. S., & Maxim, B. R. (٢٠٢٠). *Software Engineering: A Practitioner's Approach* (٩th ed.). McGraw-Hill Education.

٧. Silberschatz, A., Korth, H. F., & Sudarshan, S. (٢٠٢٠). *Database System Concepts* (٧th ed.). McGraw-Hill Education.

٨. Stallings, W. (٢٠٢٠). *Cryptography and Network Security: Principles and Practice* (٨th ed.). Pearson.

٩. Tailwind Labs. (٢٠٢٤). *Tailwind CSS Documentation: Utility-First Fundamentals*. Retrieved from .

١٠ .پدرام، حسین .١٣٩٩ .*امنیت شبکه‌های کامپیوتری و پروتکل‌های رمزنگاری* .چاپ پنجم، تهران: انتشارات ناقوس .

١١ .جهانگیری، مهدی .١٣٩٨ .*مهندسی نرم‌افزار ١: مبانی و اصول تحلیل سیستم* .تهران: انتشارات جهاد دانشگاهی صنعتی امیرکبیر .

١٢ .روحانی رانکوهی، سید محمدتقی .١٣٩٧ .*اصول طراحی پایگاه داده‌ها* .ویرایش چهارم، تهران: انتشارات جلوه .

١٣ .لطفی اربط، علی .١۴٠۴ .*طراحی و پیاده‌سازی سامانه مدیریت هوشمند و یکپارچه ساختمان (BMS) مبتنی بر وب* .پایان‌نامه کارشناسی، دانشکده برق و کامپیوتر، دانشگاه ملی مهارت تبریز .

١۴ .مرکز آمار ایران .١۴٠٢ .*گزارش وضعیت هوشمندسازی ساختمان‌های مسکونی در کلان‌شهرها* .تهران: درگاه ملی آمار .

١۵ .ملکیان، احسان .١۴٠٠ .*اصول مهندسی اینترنت و توسعه وب* .تهران: انتشارات نصر.

١۶ .صادقی، محمد؛ حیدری، سارا و شجاعی، علی. ١۴٠٣ .تحلیل و ارزیابی چالش‌های امنیتی نوین در اکوسیستم اینترنت اشیاء و راهکارهای دفاعی مبتنی بر هوش مصنوعی .فصلنامه علمی-پژوهشی پدافند سایبری و امنیت اطلاعات، سال هشتم، شماره ١.

١٧ .کریمی، امیرحسین و نوری، رضا. ١۴٠٣ .رایانش لبه‌ای (Edge Computing): پارادایم نوین در پردازش توزیع‌شده برای کاهش تأخیر در سامانه‌های بلادرنگ .تهران: انتشارات دانشگاه صنعتی شریف. (مرتبط با رویکرد پردازش محلی شما)

١٨ .احمدی، زهرا و محمدی، کاظم. ١۴٠٢ .مقایسه عملکردی پروتکل‌های لایه کاربرد در اینترنت اشیاء (CoAP, MQTT, HTTP) در محیط‌های با منابع محدود .هفتمین کنفرانس ملی پیشرفت‌های اخیر در مهندسی کامپیوتر و فناوری اطلاعات. (مرتبط با بحث مقایسه پروتکل‌ها)

١٩ .موسوی، سید علی. ١۴٠٣ .توسعه مدرن وب: راهنمای جامع پیاده‌سازی وب‌اپلیکیشن‌های پیش‌رونده (PWA) و معماری‌های میکروسرویس .تهران: انتشارات دیباگران تهران. (مرتبط با تکنولوژی فرانت‌اند و پیشنهادات آتی شما)