

انجام این تمرین به صورت انفرادی است.

دانشجویان محترم توجه داشته باشند که تنها موظف به پاسخگویی به یکی از نسخه های تمرین هستند. لذا در صورت انتخاب نسخه ی نرم افزاری (شبیه سازی) نیازی به انجام نسخه سخت افزاری نخواهد بود و یا در صورت انتخاب نسخه ی سخت افزاری نیازی به انجام نسخه ی نرم افزاری نیست.

شرح تمرین

هدف از این تمرین آشنایی با پروتکل های MQTT، COAP و HTTP و استفاده از آن ها به منظور پیاده سازی یک سناریوی واقعی اینترنت اشیاء است. در نهایت این سیستم با یک پلتفرم جهت ذخیره سازی و پردازش اطلاعات ارتباط برقرار می کند. با توجه به محدودیت های موجود در things و گره های میانی، پردازش و عملیات اصلی در پلتفرم انجام می شود.

سوالات تئوری:

۱. برتری پروتکل MQTT نسبت به پروتکل های HTTP و COAP چیست؟ توضیح دهید.
۲. بزرگترین مشکل و در واقع ضعف پروتکل MQTT چیست؟ چه راه حل هایی برای حل این مشکل پیشنهاد شده است؟ به صورت خلاصه توضیح دهید.

سوالات شبیه سازی:

شرح سناریو

در این تمرین قصد داریم یک دفتر هوشمند را پیاده سازی نماییم. در این دفتر تعدادی اتاق وجود دارد. کارمند می تواند با وارد کردن شناسه و رمز عبور خود وارد اتاق شود. در صورت تایید هویت کاربر، قفل درب باز می شود و سنسور نور با توجه به درجه روشنایی تعیین شده توسط کارمند، نور اتاق را تنظیم می کند.

در هر دفتر یک سرور محلی وجود دارد که با استفاده از پروتکل MQTT و یا COAP با نود ابتدایی در ارتباط است (در هر اتاق یک برد وظیفه دریافت اطلاعات سنسور نور و صفحه کلید، ارسال آن ها برای سرور محلی، دریافت تصمیم گیری و نهایتا کنترل عملگرها را بر عهده دارد که این برد نود ابتدایی ماست). این نود برای شما وجود خارجی ندارد در نتیجه کافی است یک برنامه cli پیاده سازی کنید که شناسه، رمز عبور و شناسه اتاق کارمند را دریافت کرده و از طریق پروتکل های مطرح شده برای سرور محلی ارسال می کند. نتیجه احراز هویت از طریق سرور محلی برای برنامه شما ارسال می شود و در صورت صحیح بودن، درب باز می شود (با نمایش پیام مناسب) و به منظور تنظیم نور، یک عدد رندوم با توجه به ساعت محلی، در صورتی که بین ساعات ۸ صبح تا ۴ بعد از ظهر قرار داشتیم بین ۶۶ تا ۱۰۰، در صورتی که بین ۸ شب تا ۴ صبح قرار داشتیم بین ۰ و ۳۳ و در غیر این صورت بین ۳۳ تا ۶۶ تولید می شود که نمایانگر میزان خوانده شده توسط سنسور است. سپس در صورتیکه مقدار تعیین شده توسط کارمند بیشتر از میزان نور بود، اختلاف این دو مقدار به عنوان روشنایی لامپ ها نمایش داده می شود (در صورت کوچکتر بودن مقدار صفر پرینت شود). کارمند با زدن دستور exit از دفتر خارج می شود (این برنامه در یک حلقه بی پایان اجرا می شود). دقت داشته باشید پروتکل ارتباطی برنامه شما با سرور محلی باید بین MQTT و یا COAP قابل انتخاب باشد.

در ادامه دو سناریو برای پیاده سازی مطرح می شود. شما موظف هستید یک مورد را انتخاب کرده و پیاده سازی نمایید. دلیل انتخاب خود را توضیح داده و tradeoff های مطرح را بیان کنید.

سناریوی اول:

سرور محلی

در این سرور یک دیتابیس وجود دارد که وظیفه ذخیره سازی کارمندان احراز هویت شده را بر عهده دارد. می توان در این سرور کاربران ادمین ایجاد کرد. این کاربران ادمین می توانند پس از احراز هویت با استفاده از نام کاربری و رمز عبور خود، با وارد کردن رمز عبور دلخواه و شناسه اتاق، کارمند جدیدی ایجاد کنند که شناسه پرسنلی آن به صورت خودکار افزایش می یابد. (این کد پرسنلی به عنوان شناسه هر کارمند جهت ورود به اتاق تعریف شده، استفاده می شود) دقت داشته باشید این شناسه باید در سرور مرکزی (که در ادامه توضیح داده می شود) یکتا باشد.

اندپوینت‌های مربوط به ادمین (موارد مشخص شده با رنگ قرمز نیاز به احراز هویت دارند):

/api/admin/login

ورود ادمین

/api/admin/register

ثبت ادمین جدید

/api/admin/user/register

ثبت کارمند جدید

/api/admin/activities

مشاهده ورود و خروج کارمندان

اندپوینت‌های مربوط به کارمند (موارد مشخص شده با رنگ قرمز نیاز به احراز هویت دارند):

/api/user/login

ورود کارمند جهت تغییر تنظیمات

کارمند می‌تواند از طریق یکی از ۲ اندپوینت زیر میزان روشنایی اتاق خود را مشخص کند (به انتخاب خود یکی را پیاده‌سازی کنید)

/api/user/:userid?lights=value

در این حالت با استفاده از query parameter تنظیمات انجام می‌شود

/api/user/:userid

در این حالت بدنه درخواست باید شامل مقدار نور باشد که از طریق آن تنظیمات انجام می‌شود

سرور مرکزی

روی سرور مرکزی باید بتوان دفتر جدیدی تعریف کرد.

/api/office/register

سرور محلی پس از احراز هویت یک کارمند، یک درخواست حاوی شناسه پرسنلی به سرور مرکزی ارسال می‌کند. سرور مرکزی پس از ثبت ورود و خروج، تنظیمات مورد نظر کارمند (شدت نور) را برای سرور محلی ارسال می‌کند.

هر سرور محلی تنها می‌تواند پس از احراز هویت به کارمندان خود دسترسی داشته باشد. این احراز هویت را توسط apikey و یا jwt پیاده‌سازی کنید.

ارتباط برد با سرور محلی از طریق پروتکل MQTT یا COAP و بقیه ارتباطات از طریق پروتکل HTTP صورت می‌گیرد.

جداول

بر روی سرور محلی تنها اطلاعات ادمین و شناسه مربوط به هر کارمند (شناسه پرسنلی) ذخیره می شود:

admins → user, password

users → id, password, room

سرور مرکزی:

users → id, password, light, office, room

activity → user, office, datetime, type

office → id, name

سرور محلی

در این سرور یک حافظه نهان داخل حافظه^۱ وجود دارد که وظیفه ذخیره سازی کارمندان احراز هویت شده را بر عهده دارد.

سرور محلی پس از دریافت اطلاعات یک کارمند، یک درخواست حاوی شناسه او به سرور مرکزی ارسال می‌کند. سرور مرکزی پس از ثبت ورود و خروج، نام و تنظیمات مورد نظر کارمند (شدت نور) را برای سرور محلی ارسال می‌کند. سرور محلی این اطلاعات را برای مدت ۱۲ ساعت در حافظه نهان خود ذخیره می‌کند و در صورت ورود مجدد، همین اطلاعات را لود می‌کند.

هر سرور محلی تنها می‌تواند پس از احراز هویت به کارمندان خود دسترسی داشته باشد. این احراز هویت را توسط apikey و یا jwt پیاده‌سازی کنید.

سرور مرکزی

روی سرور مرکزی باید بتوان دفتر جدیدی تعریف کرد.

/api/office/register

می‌توان در این سرور کاربران ادمین ایجاد کرد. این کاربران ادمین می‌توانند پس از احراز هویت با استفاده از یوزر و پسورد خود، با وارد کردن رمز عبور دلخواه، کارمند جدیدی ایجاد کنند که شناسه آن به صورت خودکار افزایش می‌یابد. دقت داشته باشید این شناسه باید در سرور مرکزی (که در ادامه توضیح داده می‌شود) یکتا باشد.

اندپوینت‌های مربوط به ادمین (موارد مشخص شده با رنگ قرمز نیاز به احراز هویت دارند):

/api/admin/login

ورود ادمین با نام کاربری، رمز عبور و شناسه دفتر

/api/admin/register

ثبت ادمین جدید

/api/admin/user/register

ثبت کارمند جدید

/api/admin/activities

مشاهده ورود و خروج کارمندان

اندپوینت‌های مربوط به کارمند (موارد مشخص شده با رنگ قرمز نیاز به احراز هویت دارند):

/api/user/login

ورود کارمند جهت تغییر تنظیمات

کارمند می‌تواند از طریق یکی از ۲ اندپوینت زیر میزان روشنایی اتاق خود را مشخص کند (به انتخاب خود یکی را پیاده‌سازی کنید)

¹ In Memory Cache

`/api/user/:userid?lights=value`

در این حالت با استفاده از query parameter تنظیمات انجام می‌شود

`/api/user/:userid`

در این حالت بدنه درخواست باید شامل مقدار نور باشد که از طریق آن تنظیمات انجام می‌شود
ارتباط برد با سرور محلی از طریق پروتکل MQTT یا COAP و بقیه ارتباطات از طریق پروتکل HTTP صورت می‌گیرد.

جداول

بر روی سرور محلی تنها اطلاعات مربوط به هر کارمند ذخیره می شود:

users → id, password, room, light

سرور مرکزی:

admins → user, password, office

users → id, password, light, office, room

activity → user, office, datetime, type

office → id, name

نحوه تحویل تمرین

۱. تحویل تمرین در قالب ۴ فایل ویدئویی انجام می‌شود که ۲ تای آن شامل سوالات تئوری، مورد سوم مقایسه سناریوهای موجود و دلیل انتخاب خود و مورد آخر بررسی کامل سناریوی پیاده‌سازی شده است. در هر ویدئو مشخص شود کدام مرحله از مراحل فوق در حال انجام است. توجه داشته باشید که در هر ویدئو تمامی مراحل کار و نتایج به طور کامل حداکثر در دو دقیقه شرح داده شود. تنها ویدیوی آخر حداکثر می‌تواند ۱۵ دقیقه باشد ولی دقت داشته باشید که باید همه موارد را به درستی پوشش دهید.
۲. در هر ویدئو باید مشخص شده باشد که این فایل متعلق به شما است. برای مثال قبل از توضیح مراحل انجام کار، یک فایل word حاوی نام فرد، شماره دانشجویی و بخش مربوطه بر روی سیستم نشان دهید که مشخص کند این ویدئو توسط شما ضبط شده است.
۳. تمرین در قالب یک فایل zip تحویل داده شود و باید برای مراحل تئوری تنها ویدیو و برای مراحل شبیه‌سازی، یک ویدئو به همراه کد وجود داشته باشد. نحوه نام‌گذاری فایل باید به صورت زیر باشد:
HW4_studentNumber.zip که در آن StudentNumber شماره دانشجویی سرگروه است. (مثال: HW4_9731079.zip)
۴. دقت کنید که حجم فایل Zip شده نهایی، حداکثر ۱۵۰ مگابایت باشد.
۵. هر مرحله از ۸ مرحله را به صورت زیر نام‌گذاری نمایید. این نحوه نام‌گذاری متناسب با تمرین خواسته شده در هر مرحله است.

01. ProtocolsComparison

02. MQTT

03. ScenarioComparison

04. ScenarioImplementation

۶. تمامی ویدئوهای ضبط شده باید قابل پخش با آخرین نسخه نرم‌افزار KMPlayer باشد.
۷. مهلت تحویل تمرین ۸ بهمن ۱۴۰۰ است.
۸. تاخیرها براساس قوانین شیوه‌نامه اعمال می‌شود.
۹. در صورت عدم رعایت موارد ذکر شده، نمره مربوط به بخش خوانایی کسر خواهد شد.

موفق و مؤید باشید