|  |  |  |
| --- | --- | --- |
| Description: Description: بازنشسته.jpg | ارتش جمهوری اسلامی ایران  (فرماندهی جنگال و سایبر راهبردی آجا ) |  |

**دفترچه کاربری**

طراحی و پیادهسازی سامانه هوشمند نظارت بر تصاویر دوربین های پیرامونی تحت شبکه

نام و نام خانوادگی مجری طرح:

حسین غلامی

زمستان 1400



**مقدمه:**

در این پروژه یک سیستم هوشمند کامل برای پیاده‌سازی در سامانه نظارتی مکان های امنیتی برای ‏نظارت پیرامونی، ارائه شد که با به‌کارگیری آن بتوان امنیت را با کمک هوش مصنوعی فراهم کنیم. بدین شکل که بتوان هر الگوریتم تشخیص شی دلخواه را بر ‏روی سیستم پیاده کرد تا کاربران اتاق مانیتورینگ بتوانند نتیجه پردازش شده این مدل­ها را روی ‏دوربین مورد نظر بررسی کنند، این کار موجب افزایش بهره­وری افراد خواهد شد، چرا که هر شخص ‏می­تواند با دقت بیشتری، تعداد دوربین بیشتری را بررسی کند و از خطای انسانی ناشی عدم دقت ‏شخص ناظر نسبت به یک نقطه جلوگیری کند

همچنین برای اینکه ‏بتوان نظارت دوربین­های مناطق متفاوت را در یک پایگاه امنیتی، در اختیار اشخاص متفاوت قرار داد و ‏از حداکثر توانمندی یک مجموعه استفاده کرد، برای هر دوربین سطح دسترسی تعریف شد، تا ‏اشخاصی که به دوربین ها دسترسی پیدا می­کنند را مدیریت کرد. به عنوان مثال در سیستم پیش رو ‏می­توان دوربین­های ورود خروج را با یک سطح دسترسی مشترک، و دوربین ها انبارهای تسلیحاتی را ‏با یک سطح دسترسی بالاتر تعریف نمود و در سیستم یک کاربر را مسئول دسترسی به دوربین های ‏ورود و خروج تعریف نمود و یک کاربر را با سطح دسترسی بالاتر تعریف نمود، با انجام این کار شخصی ‏که به دوربین های ورود خروج دسترسی دارد به اطلاعات دوربین های انبار تسلیحاتی دسترسی ‏نخواهد داشت ولی کاربری که سطح دسترسی بالاتردارد، می­تواند بر همه­ی دوربین ها نظارت داشته ‏باشد.

برای اینکه سیستم به لحاظ عملکردی پایدار باشد و رایانه­هایی که به اطلاعات دوربین­ها ‏دسترسی پیدا می­کنند، رایانه های عمومی باشند. باید پردازش اطلاعات دوربین­ها در رایانه های ‏سرویس­دهنده انجام شوند. برای کنترل این سیستم و معماری پیشنهادی به سه برنامه کاربردی نیاز ‏داریم که به ترتیب در رایانه های کاربر، رایانه سرور پردازشی، رایانه سرور اجرا شوند. در رایانه سرور ‏چند سرویس اصلی اجرا خواهد شد که هر سه برنامه­های کاربردی از آن­ها استفاده کرده و سیستم را ‏پایدار نگه خواهند داشت. این سرویس ها شامل ‏RabbitMQ‏ (وظیفه ایجاد جریان داده بین رایانه ها)، ‏MinIO‏ (وظیفه دسترسی به اطلاعات ذخیره شده) و ‏Redis‏ (وظیفه هماهنگ سازی برنامه ها و ‏کنترل دسترسی کاربران به اطلاعات) هستند، با برنامه­های کاربردی در تامل خواهند بود. برنامه ‏کاربردی سرور که بر روی رایانه سرور اجرا می­شود وظیفه تعریف دوربین جدید، فعال سازی دوربین ها ‏‏(دریافت اطلاعات از دوربین، ارسال به سرویس ‏RabbitMQ‏ و ذخیره اطلاعات) و تعیین کاربر جدید ‏را بر عهده دارد. برنامه کاربردی سرور پردازشی که بر روی رایانه سرور پردازشی اجرا می­شود وظیفه ‏دریافت اطلاعات و پردازش با الگوریتم هوش مصنوعی مورد نظر بر اساس پارامتر هایی که برای آن ‏تنظیم می­شود و برگردانند اطلاعات پردازش شده به سرویس ‏RabbitMQ‏ را بر عهده دارد. در پایان ‏نیز برنامه کاربردی کاربر که است می­تواند بر روی هر رایانه که در شبکه داخلی مجموعه قرار گرفته ‏است، اجرا شود که در این برنامه می¬توان به اطلاعات ذخیره شده دوربین ها و همچنین پخش زنده ‏دوربین ها بر اساس سطح دسترسی که برای آن کاربر مجاز است، دست یافت. رابط کاربری تمامی ‏برنامه های کاربردی با ‏pyqt‏ و زبان پایتون نوشته شد.‏

**بکارگیری و نحوه کارکرد سیستم**

برای اینکه سیستم را بتوانیم در عمل بکار بگیریم، باید پیش­نیازی هایی را آماده کنیم و سیستم را برای بکارگیری در عمل تنظیم کنیم در این مستند آماده­سازی زیرساخت های پروژه و نصب پیش­نیازها را توضیح خواهیم داد و در ادامه به آموزش تنظیمات بخش های مختلف نرم افزار های پروژه که شامل برنامه کاربردی سرور ، برنامه کاربردی سرور پردازشی و برنامه کاربردی کاربر هستند میپردازیم.

# - راه اندازی و نصب پیشنیاز ها

برای اجرای این پروژه به چند نرم افزار پایه نیاز داریم. که در ادامه به نصب آن ها میپردازیم و سرویس ها را فعال خواهیم نمود. برای اجرای این پروژه به اولین چیزی که نیاز داریم نرم افزار داکر است. که می­بایست بر روی کامپیوتر سرور نصب شود، نحوه این نرم افزار برای سرورهای ویندوزی و لینوکسی متفاوت است که هر دوی آن ها را بررسی خواهیم نمود. همچنین برای راه اندازی این پروژه به پایتون نیز نیاز داریم که در سرورهای لینوکسی به صورت پیشفرض وجود دارد ولی برای سرورهای ویندوزی به نصب آن نیاز داریم.

## نصب داکر بر روی ویندوز

برای این کار به تعدادی پیشنیاز احتیاج داریم که باید انجام شود. به طور کلی 2 شیوه برای راه­اندازی داکر در ویندوز وجود دارد که نحوه اجرای کانتینرها بر روی سخت افزار موجود را مشخص می­کند، روش اول، hyper-v backend است، که کانتینر ها توسط خود ویندوز کنترل خواهند شد. روش دوم استفاده از WSL است که کوتاه شده windows subsystem for linux است، بدین شکل که در داخل ویندوز به ماشین مجازی ای از جنس لینوکس اجرا شده و کانتینر ها در آن محیط کنترل خواهند شد. در روش دوم باید بر اساس لینک[[1]](#footnote-1) فعال سازی شود. در هر دو روش در بایوس دستگاه باید امکان شبیه­سازی[[2]](#footnote-2)فعال شده باشد و حافظه رم حداقل 4 گیگابایت و یک پردازنده 64 بیتی موجود باشد. کانتینرها و ایمیج های ایجاد شده با Docker Desktop بین تمام حساب های کاربری در ماشین­هایی که در آن نصب شده است به اشتراک گذاشته می شود. این به این دلیل است که همه حساب‌های ویندوز از یک ماشین مجازی برای ساخت و اجرای کانتینرها استفاده می‌کنند. توجه داشته باشید که هنگام استفاده از WSL امکان اشتراک گذاری کانتینرها و ایمیج ها بین حساب های کاربری وجود نداشته باشد. در ادامه باید نرم افزار Docker desktop دانلود و نصب شود.[[3]](#footnote-3)

## نصب داکر بر روی لینوکس

داکر را می­توان بر روی اکثر توزیع های لینوکس نصب نمود ولی برای نمونه، نصب داکر بر روی توزیع ubuntu شرح داده می­شود. برای اینکار تنها باید به یک پردازنده 64 بیتی و یک سیستم عامل به روز نیاز دارید. بهتر است پیش از ادامه نسخه قبلی داکر را توسط دستور زیر حذف نمایید.

$ sudo apt-get remove docker docker-engine docker.io containerd runc

برای نصب داکر در ابونتو، دو روش وجود دارد، روش اول، نصب از طریق ریپازیتوری­ها[[4]](#footnote-4) است، روش دوم گرفتن پکیج دبین آن و نصب به صورت دستی است که عموما برای کامپیوترهایی که به اینترنت دسترسی ندارند استفاده می­شود. در این بخش تنها به نصب از طریق روش اول بسنده می­کنیم. برای اینکار ابتدا پیکج apt را به روز رسانی میکنیم و پیشنیاز هایی که برای دریافت فایل نیاز دارد را ازطریق دستور زیر حل خواهیم کرد.

$ sudo apt-get update

$ sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

در ادامه کلید PGP داکر را دریافت و از طریق دستور زیر تنظیم می­کنیم.

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

و با دستور زیر یک محیط برای نصب docker engin فراهم می­کنیم. در هنگام تنظیم می­توانیم به جای کلید واژه stable از کلید واژه های nightly یا test استفاده کنیم، که برای تنظیم و انتخاب نسخه مورد دانلود است، نسخه night نسخه­ای است که هنوز منتشر نشده و روی آن کار می­کنند است.

$echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

حال با دستور زیر آخرین نسخه از docker-ce را دریافت می­کنیم.

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

در ادامه با دستور زیر لیست نسخه هایی که میتوانیم نصب کنیم را بدست می­آوریم.

$ apt-cache madison docker-ce

و از نسخه بین نسخه های موجود، آن نسخه ای که برای سیستم عامل و توزیع لینوکس کامپیوتر میزبان مناسب است را انتخاب میکنیم و در پایان نصب docker-engine را نصب می­کنیم.

$ sudo apt-get install docker-ce=<VERSION\_STRING> docker-ce-cli=<VERSION\_STRING> containerd.io

## نصب پایتون

برنامه های کاربردی در این پروژه تماما با پایتون 3.8 توسعه داده شده است، از آن جایی که در ویندوز این برنامه به صورت پیش­فرض وجود ندارد، می­بایست آن را نصب نماییم. برای اینکار به لینک[[5]](#footnote-5) رفته و فایل مربوطه دانلود و نصب می­کنیم.

### نصب کتابخانه ها و ایمیج های استاندارد

1-برای این کار به دایرکتوری اصلی پروژه که شامل محتوای زیر است بروید.

‎├── Client application

├── Data

├── Server

├── Server\_process layer

├── docker-compose.yml

├── env

├── rbmq

├── redis

└── requirements.txt

2-دستور زیر را در خط فرمان وارد کنید تا کتابخانه های مورد نیاز پایتون نصب شوند.

pip install requirements.txt

3- دستور زیر را در خط فرمان وارد کنید تا ایمیج های مورد نیاز داکر دریافت شوند.

docker-compose build

# راهنمای استفاده کاربران

در بخش قبل پیش نیاز های پروژه را دریافت و نصب نمودیم حال وقت آن رسیده که سرویس ها را فعال کرده و یک از برنامه ها را اجرا کنیم. در این بخش راهنمای استفاده از هر یک از برنامه کاربردی سرور، برنامه کاربردی سرور پردازشی و برنامه کاربردی کاربر را بررسی خواهیم نمود.

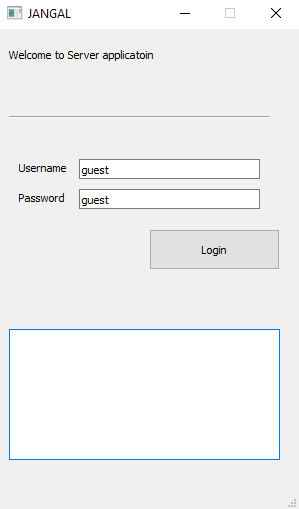
## راهنمای راه اندازی برنامه کاربردی سرور

پیش از آن که برنامه کاربردی سرور را اجرا کنیم نیاز داریم که سرویس­های مورد استفاده پروژه را که در بخش قبل دریافت نمودیم فعال کنیم. برای اینکار کافی است در مسیر اصلی پروژه دستور زیر را وارد نماییم.

docker-compose up

با بالا آمدن سرویس­های پروژه تنها کافی است برنامه کاربردی سرور را اجرا کنیم، برای اینکار وارد پوشه سرور شده و برنامه server.py را با دستور زیر به اجرا در بیاورید.

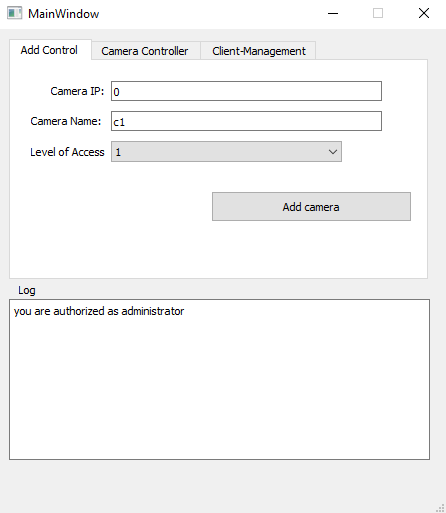
python server.py

با اجرای این دستور فوق برنامه گرافیکی که در شکل ‏1–1 مشاهده می­کنید، به اجرا در خواهد آمد. در این صفحه از شما رمز عبور و نام کاربری سرور را می­خواهد. این نام کاربری و رمز عبور در حقیقت، شناسه کاربری کاربر مدیرسیستم در سرویس rabbitmq است. که با وارد شدن در پنل وب این سرویس و می­توان آن را تغییر داد. (جزئیات بیشتر در پیوست آمده است.)

این نام کاربری و رمز عبور به صورت پیش­فرض guest و guest است. در ادامه با فشردن کلید login میتوان وارد این نرم افزار شد. خطا هایی که این نرم افزار می­تواند بدهد در پایین صفحه نوشته خواهد شد که به شرح زیر است: خطای ‏authorizatoin fail -->not\_authorized‎‏ به معنای آن است که شناسه کاربری یا رمز عبور اشتباه است.‏ خطای ‏authorizatoin fail -->offline‎‏ به معنای آن است که نرم افزار داکر اجرا نشده است.

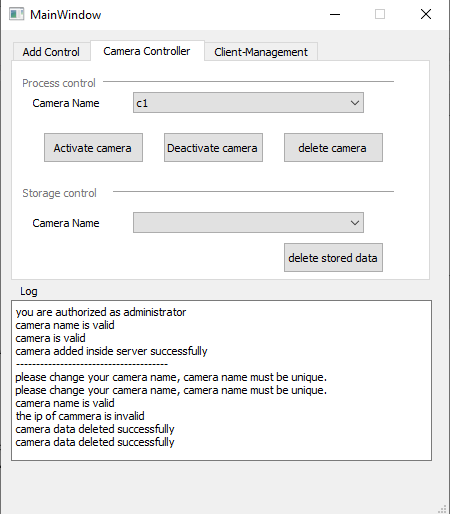
شکل 1–1-صفحه ورود برنامه کاربردی سرور

با وارد شدن به برنامه کاربردی سرور، صفحه لاگین بسته شده و صفحه ای که در شکل 1–2 مشاهده می­کنید رو به رو خواهید شد.‏ این نرم افزار از چند تب‏ تشکیل شده است. که شامل تب اضافه کردن دوربین جدید به سیستم، تب کنترل کردن دوربین های سیستم، تب کنترل کاربران سیستم است.

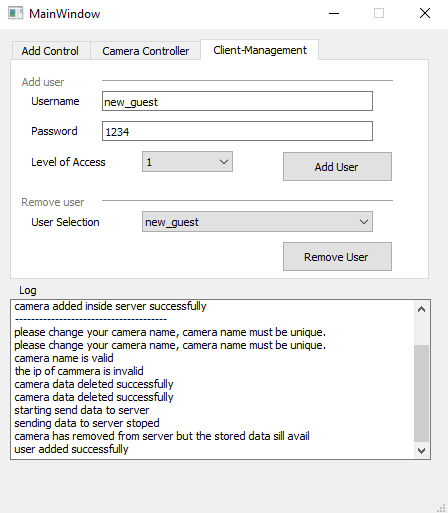
با تب اضافه کردن دوربین جدید به سیستم شروع می­کنیم. در خط اول لینک ‏rtsp‏ دوربین مربوطه را وارد شود. در خط دوم نامی منحصر به فرد برای دوربین تعیین شود. سطح دسترسی به دوربین مربوطه را مشخص شود. کلید اضافه کردن دوربین جدید فشرده شود. با فشردن این کلید، یک مرکز توزیع داده بر اساس نامی که برای دوربین انتخاب کرده اید در بروکر تعریف می­شود و بستری برای اجرای فرآیند فراهم می­شود.

حال در این میان خطاهایی می­توانند ایجاد شوند. خطای ‏please change your camera name, camera name must be unique‏ به ‏معنای آن است که نام مورد نظر استفاده شده و باید نام دیگری برای دوربین انتخاب کنید.‏ خطای ‏the ip of cammera is invalid‏ به معنی این است که لینک دسترسی به دوربین اشتباه ‏وارد شده است.‏ در صورت صحیح بودن همه فیلد ها ‏camera added inside server successfully‏ چاپ می­شود.

شکل 1–2- تب اول از برنامه کاربردی سرور-اضافه کردن دوربین جدید

حال به سراغ تب دوم خواهیم رفت، برای این کار کافی است روی آن کلیک کنیم در شکل ‏1–3 میتوانیم محتوای تب دوم یعنی کنترل دوربین ها را مشاهده کنیم. این تب از دو بخش اصلی تشکیل شده است، بخش اول کنترل فرآیند دوربین­ها است و بخش دوم، کنترل اطلاعات ذخیره شده است. در بخش اول یک لیست انتخابی داریم که در آن دوربین هایی که در تب اول تعریف شده اند قرار میگیرند، با انتخاب دوربین مورد نظر می­توانیم هر یک از سه کلید زیر آن را بفشاریم. کلید اول از سمت چپ برای فعال کردن فرآیند دریافت اطلاعات از دوربین ، ارسال به سرور و ذخیره اطلاعات است. کلید دوم برای غیر فعال کردن فرآیند فعال شده است. کلید سوم برای حذف دوربین است. در صورتی که دوربین انتخابی را حذف کنید باید مجدد در تب اول آن را تعریف کنید. با فشردن این کلید اطلاعات ذخیره شده دوربین حذف نخواهند شد. و تنها جایگاه فرآیند آن حذف می­شود. این بدان معناست که میتوان مجددا دوربینی با این نام در صفحه قبل ایجاد کرد و هنگام ساخت دوربین جدید با خطای please change your camera name, camera name must be unique مواجه نخواهید شد. با فشردن کلید اول در صورتی که موفقیت آمیز باشد نوشتار starting send data to server چاپ خواهد شد. با فشردن کلید دوم نوشتار sending data to server stoped‏ چاپ خواهد شد. در بخش دوم این تب یعنی کنترل اطلاعات ذخیره شده بر روی سرور، می­توان اطلاعاتی که از دوربین ها روی سرور ذخیره شده اند را حذف نمود. بدین شکل در لیست انتخابی دوربین هایی که از آن ها در سرور اطلاعات وجود دارد قرار خواهند گرفت، با انتخاب دوربین مورد نظر و فشردن کلید حذف اطلاعات می­توان این اطلاعات را حذف نمود. اطلاعات ذخیره شده در سرور در پوشه Data در مسیر اصلی پروژه قرار دارند، که می­توان به صورت مستقیم به آن اطلاعات دسترسی پیدا کرد. این پوشه در حقیقت به پوشه data که داخل کانتینر MinIO قرار گرفته است، متصل شده و با تغییر(حذف و اضافه کردن) محتوای این پوشه محتوای داخل کانتینر تغییر می­کند. پس کلید حذف اطلاعات در حقیقت محتوای دوربین انتخابی از این پوشه را پاک می­کند.حال به سراغ تب سوم یعنی، تب کنترل کاربران می­رویم، این تب را می­توان در فشردن نام آن باز کرد و در شکل ‏1–4 می­توانیم محتوای آن را مشاهده کنیم. در این تب می­توان کاربران جدیدی در سیستم ثبت نمود. که به اطلاعات دوربین ها ‏دسترسی پیدا کنند. این بخش از دو بخش اضافه کردن کاربر جدید و حذف کاربر موجود تشکیل شده است.‏ در بخش اضافه کردن کاربر جدید می­توان شناسه کاربری و رمز عبور را برای کاربر تعیین نمود و سطح ‏دسترسی آن کاربر را از 1 تا 4 مشخص نمود و کلید اضافه کردن کاربر جدید را فشرد با این کار در redis در بخش acl یک کاربر جدید تعریف می­شود که، برنامه کاربردی کاربر می­تواند با این نام کاربری و رمز عبور بر اساس سطح دسترسی­ای که برایش تعریف شد وارد برنامه شود. در بخش حذف کاربر نیز می­توان کاربران اضافه شده به سیستم را از طریق ‏لیست انتخابی از کاربران فعال‏ انتخاب کردو و با ‏فشردن کلید حذف کاربر آن ها را‏ از سیستم حذف نمود.‏ در صورتی که کاربر با موفقیت به سیستم اضافه شود، user added successfully چاپ خواهد شد و در صورت حذف کاربر از سیستم user removed successfully چاپ خواهد شد.

شکل 1–3- تب دوم از برنامه کاربردی سرور- کنترل دوربین­ها

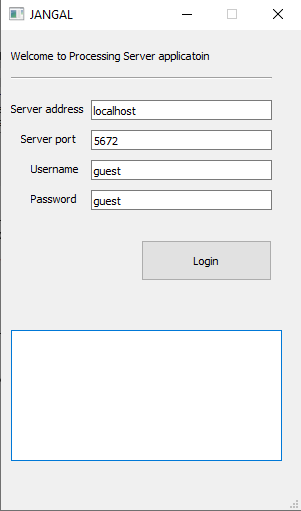


شکل ‏1–4-تب سوم از برنامه کاربردی سرور-کنترل کاربران

## راهنمای راه اندازی برنامه کاربردی سرور پردازشی

این برنامه را می­توان بر روی کامپیوتری مستقل از کامپیوتر سرور اصلی اجرا کرد. این سیستم تنها نیاز به پایتون دارد. و برای اجرای آن روی یک سیستم دیگر کافی است که کتابخانه های پایتونی که در فایل ‏requaetment.txt‏ قرار گرفته اند را نصب کرد. برای اجرای این برنامه کافی است که پوشه ‏serverprovcessor ‎‏ را به سیستم مد نظر منتقل نمود و وارد آن شد و با اجرای دستور زیر این برنامه را اجرا کرد.

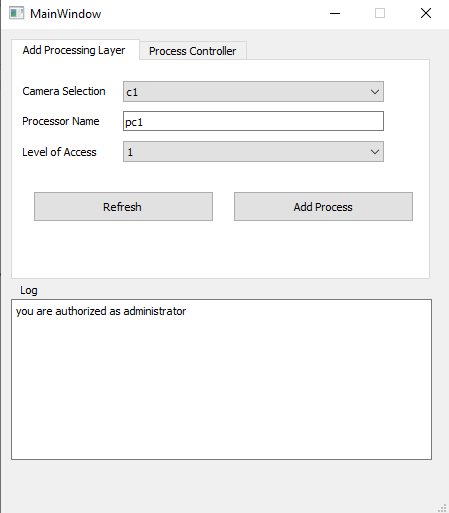
python server.py

در شکل ‏1–5 می­توان صفحه لاگین برنامه کاربردی سرور پردازشی را مشاهده کرد. که پس اجرای این برنامه نمایش داده خواهد شد. در خط اول این برنامه باید آدرس ip کامیپوتری که برنامه کاربردی سرور در آن اجرا شده است را وارد کنید. در صورتی که این برنامه را بر روی همان سیستم اجرا میکنید کافی است که ip، 127.0.0.1 یا عبارت localhost را وارد کنید

در خط بعدی پورت مربوط به سرویس rabbitmq را وارد کنید، که به صورت دیفالت 5672است. مگر اینکه در داخل داکر کامپوز تغییر یابد یا بر اساس تنظیمات شبکه پورت مذکور بر روی پورت دیگری منتقل شود. نام کاربری و رمز عبور نیز، مشابه برنامه کاربردی سرور guest , guest ، یعنی کاربر پیش فرض بروکر است. با تنظیم مقادیر فوق می­توان کلید ورود را فشرد و وارد نرم افزار سرور پردازشی شد. در صورتی که آدرس سرور به درستی وارد نشود با خطای rabbit\_not\_found مواجه خواهید شد. یا در صورت اشتباه بودن نام کاربری و رمز عبور با خطای authorizatoin fail -->not\_authorized رو به رو خواهید شد.

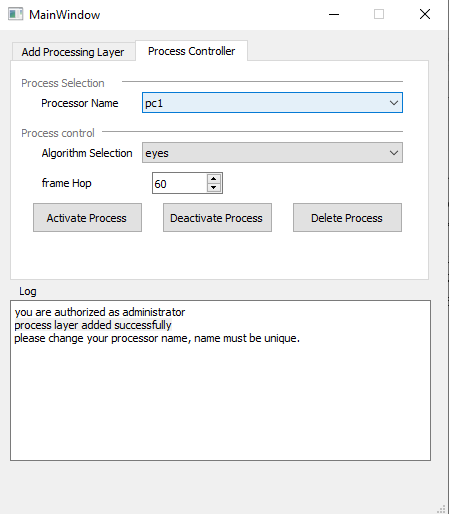
شکل ‏1–5-صفحه لاگین برنامه کاربردی سرور پردازشی

در صورتی که بدون خطا وارد سیستم شوید عبارت you are authorized as administrator چاپ خواهد شد و این صفحه بسته شده و صفحه شکل ‏1–6 نمایان خواهد شد.

این برنامه کاربردی از دو تب تشکیل شده که فرآیند پردازش بسته ها را به عهده دارند. تب اول برای اضافه کردن لایه پردازشی جدید و تب دوم برای کنترل لایه پردازشی اضافه شده است. ابتدا به سراغ تب اول یعنی اضافه کردن لایه پردازشی جدید می­رویم. در این تب در لیست انتخابی اول می­توان به دوبین­های فعال شده بر روی برنامه کاربردی سرور دسترسی پیدا کرد، در پایین این صفحه یک کلید بازیابی وجود دارد تا این برنامه از دوربین های فعال شده جدید روی سیستم مطلع شوند. ‏ پیشنهاد می­شود قبل از اضافه کردن لایه پردازشی جدید ابتدا کلید ‏refresh‏ را فشار دهید تا از آخرین وضعیت سرور ‏اطلاع کسب کنید.

در ادامه نیاز است که نامی منحصر به فرد برای لایه پرداشی مد نظر اعمال کنید و در آخرین گزینه از لیست انتخابی موجود ‏سطح دست رسی برای پردازش را انتخاب شود و سپس بر روی کلید اضافه کردن لایه پردازشی فشار دهید. لازم به ذکر است که در صورتی که سطح دست رسی یک دوربین بالاتر باشد نمی­توان لایه پردازشی را برای ‏سطوح پایین تر انتخاب نمود. به عنوان مثال اگر سطح دسترسی یک دوربین 3 باشد نمی توان سطح دسترسی 1 و 2 را ‏برای آن اعمال نمود[[6]](#footnote-6). با فشردن کلید اضافه کردن لایه پردازشی جدید خطای زیر ممکن است رخ دهند. خطای please change your processor name, name must be unique.‎‏ خطای یکسان بودن نام لایه پردازشی است و کاربر باید نام لایه پردازشی را تغییر دهد. و در صورتی که عملیات اضافه کردن لایه موفقیت انجام شود process layer added successfully‏ چاپ خواهد شد. با اینکار یک مرکز توزیع داده با نامی که در نظر میگیرید بر روی بروکر پیام آماده خواهد شد تا اطلاعات پردازش شده به آن منتقل شود.

شکل ‏4–6-تب اول برنامه کاربردی سرور پردازشی – اضافه کردن لایه پردازشی جدید

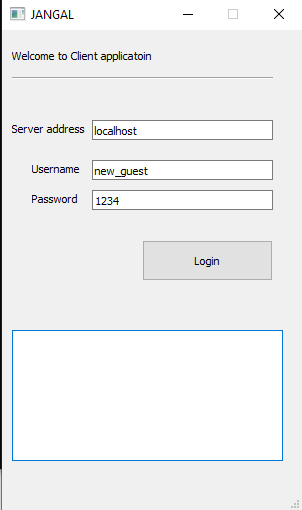
حال به سراغ تب بعدی، یعنی تب کنترل لایه پردازشی می­رویم نمای کلی این تب را می­توان در شکل زیر مشاهده نمود.‏‎ ‎ برای وارد شدن به تب کافی است بر روی نام کلیک کنید. این تب از دو بخش تشکیل شده است. بخش اول انتخاب لایه پردازشی مورد نظر است که در تب اول تعریف شد و ‏بخش دوم امر کنترل آن است که می­توان مدل هوش مصنوعی مورد نظر را انتخاب نمود.‏ گزینه بعدی یعنی پرش فریم،‏ کنترل تعداد فریم هایی است که قرار است پردازش شوند و انتخاب این گزینه به این معنا است که از هرn فریم، تنها یک فریم ‏پردازش می­شود. که بسته به مدل هوش مصنوعی و زمان پردازش هر فریم آن باید کنترل شود. چرا که زمان اجرای هر ‏الگوریتم هوش مصنوعی بر روی یک فریم از تصویر با الگوریتم دیگری متفاوت است و بستگی به سیستم در حال ‏محاسبه و تعداد پردازش هایی که بر روی آن سیستم دارد.‏ به طور کلی دوربین ها با 30 فریم بر ثانیه تصویر برداری میکنند. (در تنظیمات داخل دوربین قابل تعیین است) ‏پس می­توان گفت وقتی این پارامتر را بر روی 60 تعیین می­کنیم از هر 60 فریم یک فریم پردازش می­شود، 60 فریم در این مثال به معنای دو ثانیه است، که یعنی هر دو ثانیه یک فریم پردازش شود. برای شروع فرآیند پردازش نیز با فشردن کلید ‏فعال­سازی فرآیند‏ می­توان این لایه را فعال نمود. با فعال نمودن این لایه کاربر قادر خواهد بود به ‏اطلاعات پردازش شده دوربین دسترسی پیدا کند و با فشردن کلید ‏غیرفعال­سازی‏ فرآیند، فرآیند پردازش متوقف ‏خواهد شد. همچنین با فشردن کلید حذف‏ لایه پردازشی، این لایه حذف شده و نام آن آزاد می­شود و مرکز توزیع داده حذف می­شود و جایگاه اجرای فرآیند حذف خواهد شد.

شکل ‏1–7-تب دوم از برنامه کاربردی سرور پردازشی – کنترل لایه پردازشی

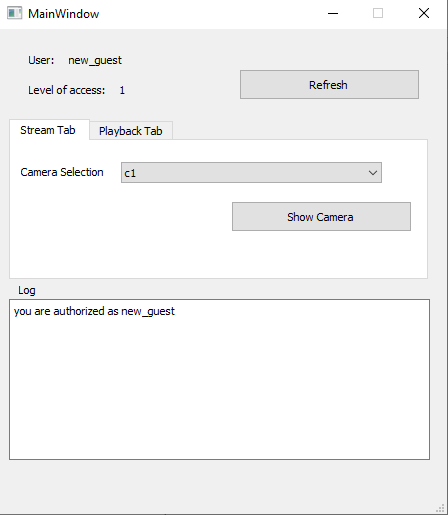
### راهنمای راه اندازی برنامه کاربردی کاربر

این برنامه را می­توان بر روی کامپیوتری در شبکه داخلی کامپیوتر سرور اجرا کرد. این سیستم تنها نیاز به پایتون دارد. و برای اجرای آن روی یک سیستم دیگر کافی است که کتابخانه های پایتونی که در فایل ‏requaetment.txt‏ قرار گرفته اند را نصب کرد. برای اجرای این برنامه کافی است که پوشه ‏client ‎‏ را به سیستم مد نظر منتقل نمود و وارد آن شد و با اجرای دستور زیر این برنامه را اجرا کرد.

python client.py

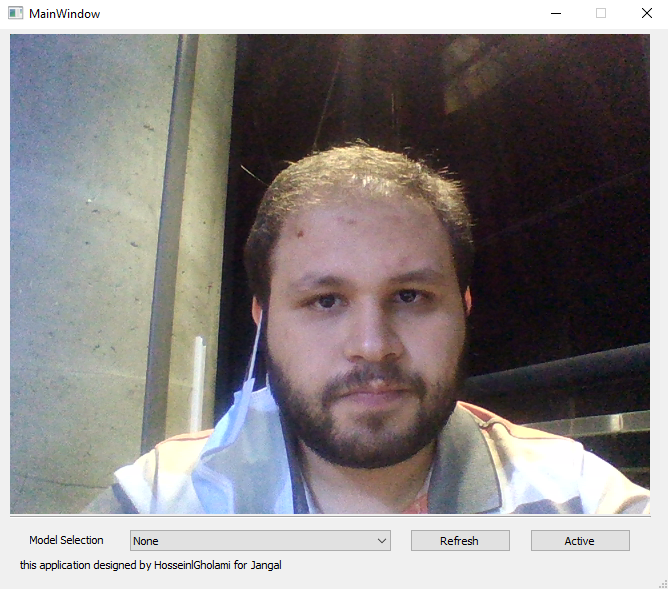
برنامه کاربردی کاربر برای دسترسی به اطلاعات دوربین ها، اطلاعات پردازش شده دوربین ها و بازپخش و دانلود اطلاعات ذخیره شده دوربین ها طراحی شده است. و براساس نام کاربری و رمز عبوری که کاربر وارد می­کند، توسط سیستم شناسایی خواهد شد و اطلاعات دوبین های مرتبط با خودش را به وی نمایش خواهد داد. با اجرای دستور فوق برنامه گرافیکی شکل 1–8 به کاربر نمایش داده خواهد شد که در خط اول ip کامپیوتری که برنامه کاربردی سرور در آن اجرا شده است را وارد کنید.

شکل ‏1–8-صفحه لاگین برنامه کاربردی کاربر

در این برنامه ابتدا ip کامپیوتری که برنامه سرور بر روی آن اجرا شده است را وارد کنید و در خط بعدی با نام کاربری و گذرواژه ای که در اپلیکیشن سرور تعیین شد میتوان وارد اپلیکیشن شد. در صورتی که نام کاربری و رمز عبور اشتباه باشد با خطای authorizatoin fail -->not\_authorized رو به روی خواهید شد. و در صورتی که نام کاربری و رمز عبور درست باشد، عبارت you are authorized as USERNAME چاپ خواهد شد.

شکل 1–9-تب اول برنامه کاربردی کاربر- پخش زنده اطلاعات

سپس این صفحه بسته شده و صفحه رابط کاربری کاربر نمایش داده خواهد شد. این صفحه را می­توانید در شکل ‏1–9 مشاهده کنید. در این صفحه در بخش بالا نام کاربری و سطح دسترسی کاربر نمایش داده می­شود. در کنار آن یک کلید بازیابی وجود دارد تا اطلاعاتی که در تب ها هستند را به روزرسانی کند. این برنامه دو وظیفه اصلی پخش زنده اطلاعات و باز پخش اطلاعات را بر عهده دارد، که این امکان در دو تب تعبیه شده است. ابتدا به بررسی تب اول می­پردازیم در این تب یک لیست انتخابی وجود دارد که در آن لیست دوربین های فعال در برنامه کاربردی سرور نمایش داده می­شود. با انتخاب از این لیست به سادگی میتوان فرآیند مربوط به پخش زنده اطلاعات را فعال کرد و به اطلاعات زنده دوربین دسترسی پیدا کرد. در صورتی که فرآیند پخش زنده به درستی اجرا شود، show camera :CAMERA\_NAMEبه نمایش خواهد آمد و در صورتی که این پنجره برای یک دوربین از قبل در حال اجرا باشد، برای مدیریت منابع اجازه اجرا داده نخواهد شد و خطای this cam window is already open, close that first بر روی لاگ ثبت خواهد شد. با فعال شدن این فرآیند در شکل ‏1–10 می­توان این پنجره را مشاهده نمود که در یک پنجره که به صورت مجزا باز می­شود، اطلاعات زنده دوربین انتخابی به نمایش گذاشته می­شود. همانطور که قابل مشاهده است در پایین این پنجره دو کلید و یک لیست انتخابی وجود دارد که وظیفه دریافت اطلاعات پردازش شده را بر عهده دارند. یک کلید بازیابی وجود دارد تا از الگوریتم های هوش­مصنوعی که پس از باز شدن پنجره برای این دوربین فعال شده اند با خبر شد و لیست انتخابی به روز شود. همانطور که در شکل1 –11 مشاهده می­کنید با فشردن بر روی لیست انتخابی، الگوریتم های هوش مصنوعی که برای این دوربین فعال هستند به نمایش در می­آید.

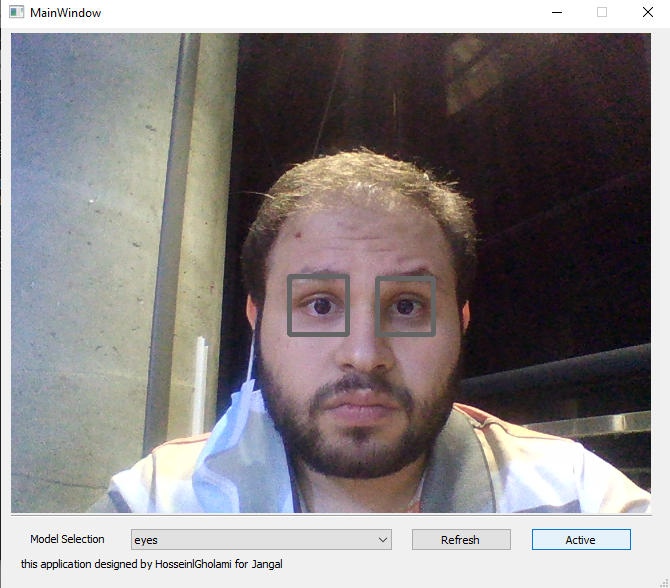


شکل ‏1–10پنجره پخش زنده اطلاعات دوربین

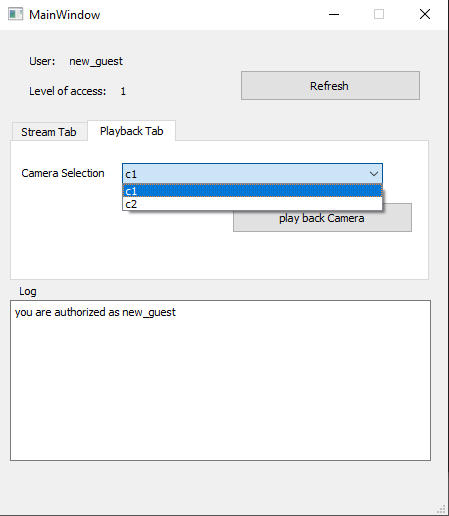


شکل ‏1–11-لیست انتخابی از الگوریتم های هوش مصنوعی فعال شده برای دوربین باز شده.

حال می­توان مدل هوش­مصنوعی مدنظر را انتخاب نمود و فشردن کلید فعال­سازی به اطلاعات این مدل دسترسی پیدا کرد. برای مثال با انتخاب مدل eyes و فشردن کلید فعال سازی این مدل بر روی ویدیو به نمایش خواهد آمد. در تصویر زیر میتوان نمونه آن را مشاهده نمود. از آن جایی که این الگوریتم را ما برای 60 فریم تنظیم کردیم و اطلاعات دوربین با نرخ 30 فریم بر ثانیه می­آید، هر دو ثانیه یک فریم پردازش خواهد شد، و جای چشم ها را تشخیص می­دهد. همچنین برای غیر فعال کردن نمایش اطلاعات پردازش شده کافی است در لیست انتخابی بر روی گزینه None کلیک کرده و سپس بر کلید فعال­سازی را انتخاب کنیم.

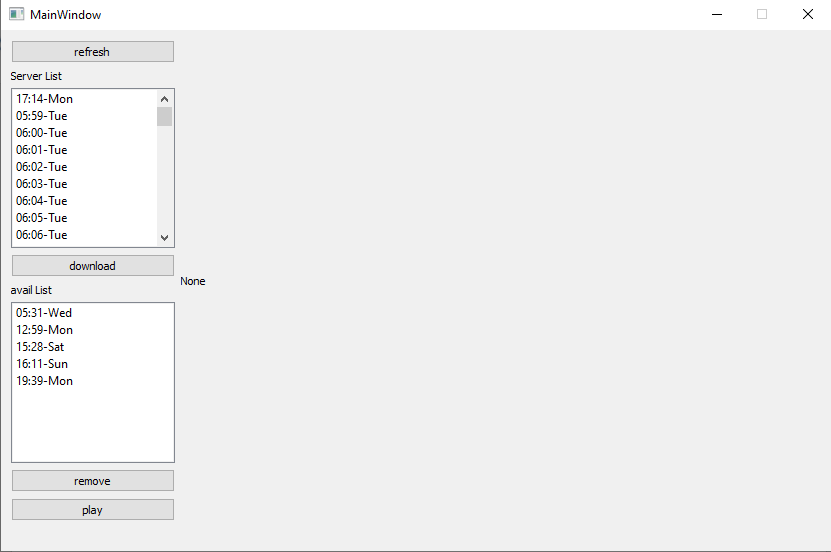


شکل ‏1–12-نتیجه فعال سازی الگوریتم تشخیص چشم بر روی دوربین

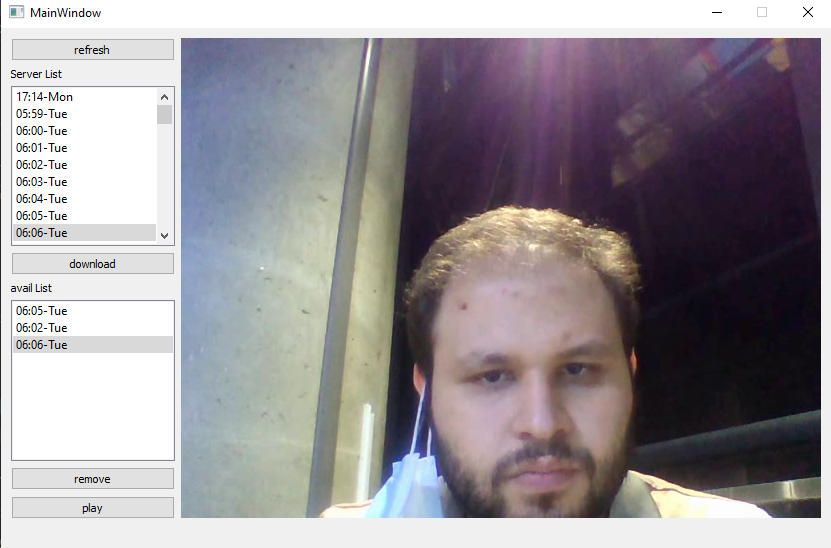
حال به سراغ تب دوم برنامه کاربردی کاربر یعنی باز پخش اطلاعات ذخیره شده میرویم در شکل ‏1–13 می­توانیم محتوایات این تب را مشاهده کنیم. در این تب یک لیست انتخابی وجود دارد که دوربین­هایی که از آن ها در سرور اطلاعات داریم را به نمایش میگذارد. همچین یک کلید وجود دارد که با انتخاب دوربین مدنظر فرآیند بازپخش اطلاعات، برای دوربین انتخابی فعال می­شود.

با انتخاب این گزینه رابط کاربری بازپخش اطلاعات ذخیره شده به عنوان یک پنجره جدید باز می­شود که در شکل ‏1–14 قابل مشاهده است. این پنجره از دو بخش اصلی تشکیل شده، یه بخش برای دسترسی به اطلاعات روی سرور، که یک کلید بازیابی برای دسترسی به آخرین اطلاعات ذخیره شده روی سرور وجود دارد، یک لیست انتخابی برای نمایش اطلاعاتی که روی سرور وجود دارد که بر اساس روز هفته و زمان اطلاعات روی سرور را نمایش می­دهد، و یک کلید دریافت اطلاعات که فایل انتخابی که بر روی سرور است را به کامپیوتر کابر منتقل می­کند. بخش دوم این برنامه از یک لیست که شامل اطلاعاتی است که از سرور بر روی کامپیوتر کاربر دریافت شده تشکیل می­دهد که با دو کلید که در زیر آن قرار گرفته مرتبط است، کلید اول کلید حذف است، بدین معنا که گزینه انتخابی از لیست اطلاعات موجود در کامپیوتر کاربر را حذف می­نماید و کلید بعدی کلید پخش اطلاعات است که در سمت راست تصویر فایل یک دقیقه دریافت شده را به نمایش می­گذارد. در شکل ‏1–15می­توان نحوه نمایش اطلاعات را مشاهده نمود. همچین لازم به ذکر است که اطلاعات دریافت شده در پوشه ای به نام download که در مسیری برنامه کاربر اجرا شده است. قرار خواهند گرفت و کابر با هر برنامه پخش ویدویی که مد نظر داشته باشد میتواند به اطلاعات دوربین ها دسترسی پیدا کند.

شکل -113-تب دوم برنامه کاربردی کارب-بازپخش اطلاعات ذخیره­شده



شکل ‏1–14-پنجره بازپخش اطلاعات ذخیره شده دوربین



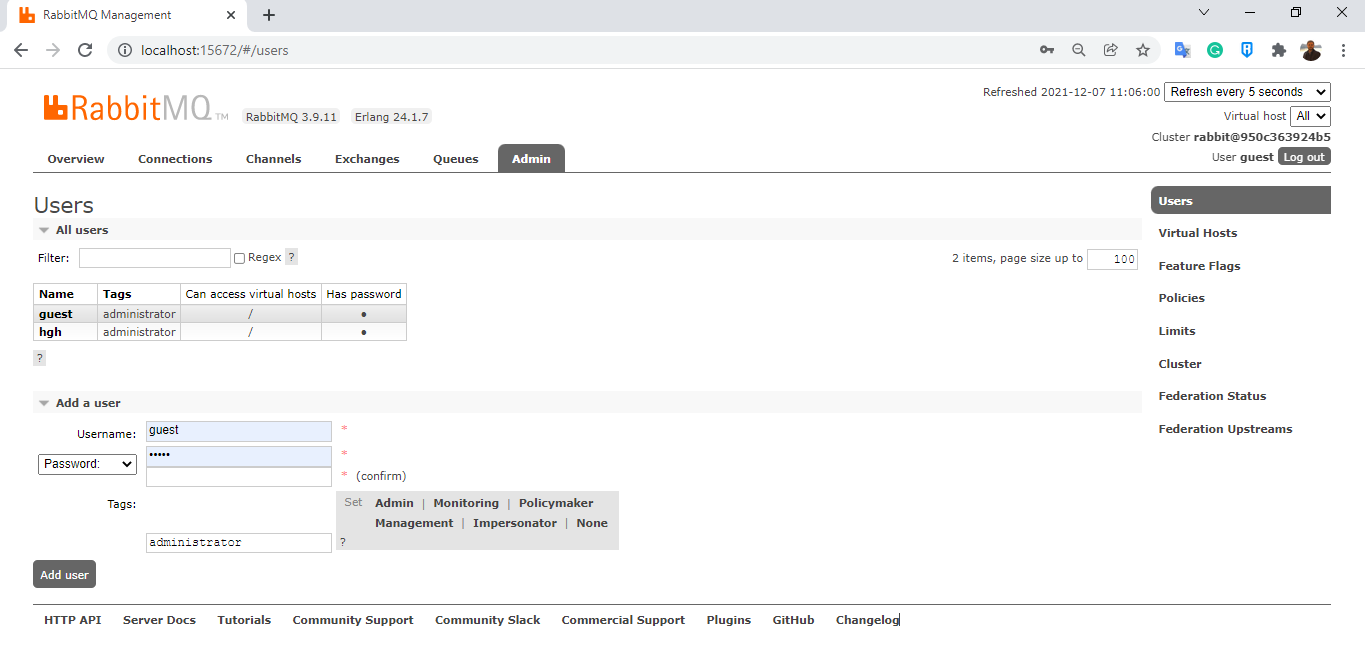
شکل ‏1–15- پخش اطلاعات ذخیره شده در پنجره بازپخش اطلاعات ذخیره شده

**پیوست 1- تغییر نام کاربری و گذرواژه اپلیکیشن های سرور**

بعد از اینکه نرم افزارهای مربوط به داکر بالا آمد، بر روی کامپیوتر میزبان کافی است به مانیور وب rabbitmq مراجعه کنید. برای این کار بر روی کامپیوتر سرور مرورگر خود را باز کرده و به سرویس rabbit بر روی پورت 15672 متصل شوید.

<http://localhost:15672/>

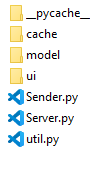
و با نام کاربری و گذرواژه دیفالت Login کنید.

از منوهای بالا وارد تب ادمین شده:  


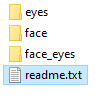
و کاربر guest را حذف نموده و کاربر جدید خود را با دسترسی admin تعریف کنید.

**پیوست 2-اضافه کردن مدل هوش مصنوعی جدید به پروژه**

برای این کار ابتدا وارد پوشه Server\_process layer شوید.



در داخل پوشه model مدل های هوش مصنوعی قرارگرفته اند.



ابتدا مدل خود را در این پوشه قرار دهید و مطابق فایل readme عمل کنید

بدین نحوه که

def get\_object\_position(frame,loc):

تابعی با این نام و آرگومان برای استفاده از مدل خود تعریف کنید. که frame فریم تصویر ورودی به مدل است و loc ادرس فایلی است که مدل شما به آن نیاز دارد. و در دایرکتوری داخل مدل قراردارد.

این تابع باید خروجی ای متشکل از یک لیست که محتوای نمونه های آن تاپل هایی هستند که حاوی مختصات های گوشه تصویر را شامل میشود باشد. ترتیب قرارگیری این متخصات ها معادل api ترسیم مستطیل در opencv است. چراکه در برنامه سمت کاربر از این api استفاده شده است.

حال که مدل به پروژه اضافه شد، کافی است در فایل sender.py این تابع در کنار سایر مدل ها import شود.

if ALGORITHM =='face':

    from model.face.object\_detection import get\_object\_position

elif ALGORITHM =='eyes':

    from model.eyes.object\_detection  import get\_object\_position

elif ALGORITHM =='face\_eyes':

    from model.face\_eyes.object\_detection  import get\_object\_position

#elif ALGORITHM =='YOUR\_FOLDER\_NAME':

    # from model.'YOUR\_FOLDER\_NAME'.object\_detection  import get\_object\_position

1. [*https://docs.microsoft.com/en-us/windows/wsl/install*](https://docs.microsoft.com/en-us/windows/wsl/install) [↑](#footnote-ref-1)
2. virtualization [↑](#footnote-ref-2)
3. [*https://docs.docker.com/desktop/windows/install*](https://docs.docker.com/desktop/windows/install) [↑](#footnote-ref-3)
4. Install using the repository [↑](#footnote-ref-4)
5. [*https://docs.docker.com/desktop/windows/install*](https://docs.docker.com/desktop/windows/install) [↑](#footnote-ref-5)
6. نمایش داده نخواهد شد. [↑](#footnote-ref-6)