

پروژه Image Authenticator با هدف ایجاد یک سیستم امن و کارآمد برای احراز هویت کاربران بر اساس تشخیص چهره است. در این سیستم از فناوری‌های مختلفی از جمله Go برای ایجاد API و Python برای پردازش تصویر و ادغام‌های API خارجی استفاده می‌کند. این پروژه ذخیره ایمن تصاویر کاربر، تشخیص چهره دقیق و ارتباط یکپارچه بین اجزا را تضمین می‌کند.

Go مسئول ثبت نام کاربر، ذخیره سازی تصویر، تعاملات پایگاه داده و ارتباط با سرویس‌های خارجی است.

Golang: به دلیل عملکرد بسیار سریع و سبک بودن ایجاد یک RESTful API از این زبان استفاده شده است.

Echo: یک چارچوب وب سبک و سریع Go برای ساخت API های RESTful است.
PostgreSQL: به عنوان سیستم پایگاه داده برای ذخیره داده‌های کاربر و وضعیت‌های احراز هویت استفاده می‌شود.

RabbitMQ: به عنوان یک صف پیام برای ارتباطات بین سرویس و پردازش مبتنی بر صف استفاده می‌شود.

ثبت نام کاربر: ثبت نام کاربر را کنترل می‌کند، داده‌های ورودی را تأیید می‌کند و تصاویر را در فضای ذخیره سازی ایمن ذخیره می‌کند (Amazon S3).

تشخیص چهره: از یک API خارجی برای شناسایی و مقایسه چهره‌ها در تصاویر برای احراز هویت کاربر استفاده می‌کند.

تعاملات پایگاه داده: عملیات CRUD را در پایگاه داده PostgreSQL برای مدیریت داده‌های کاربر و وضعیت‌های احراز هویت انجام می‌دهد.

پردازش پیام: RabbitMQ را برای پردازش پیام‌ها و انجام اقدامات بر اساس نتایج تشخیص چهره یکپارچه می‌کند.

Handlers: حاوی نقاط پایانی API برای ثبت نام کاربر، بررسی وضعیت و پردازش پیام است.

Datastore: تعاملات با پایگاه داده PostgreSQL و Amazon S3 را برای ذخیره سازی تصویر مدیریت می‌کند.

RabbitMQ: اتصالات، انتشار پیام و مصرف را با RabbitMQ مدیریت می‌کند.

پردازش تصویر (ماژول‌های پایتون)

ماژول‌های پایتون وظایف پردازش تصویر از جمله تشخیص ویژگی‌های چهره، تشخیص چهره و ارتباط با API های خارجی را بر عهده دارند.

فن آوری های مورد استفاده:

Python: به دلیل سادگی و کتابخانه‌های گسترده برای پردازش تصویر و ادغام API انتخاب

شده است.

Pillow: برای کارهای تغییر تصویر و تبدیل تصاویر به فرمت باینری استفاده می‌شود.

Requests: برای ایجاد درخواست های HTTP به API های خارجی استفاده می‌شود.

Imagga API: یک سرویس خارجی که برای تشخیص میزان شباهت و تشخیص ویژگی‌های

چهره استفاده می‌شود.

MailGun: برای ارسال ایمیل به کاربران استفاده می‌شود.

تشخیص ویژگی چهره: با استفاده از Imagga API ویژگی‌های چهره را از تصاویر استخراج

می‌کند.

تشخیص شباهت چهره: ویژگی‌های چهره دو تصویر را برای تعیین شباهت با هم مقایسه می‌کند.

ارسال ایمیل: اعلان های ایمیل را بر اساس نتایج احراز هویت برای کاربران ارسال می‌کند.

ساختار کد:

Face Detection: شامل توابعی برای تعامل با Imagga API برای استخراج ویژگی‌های

صورت.

Face Similarity: الگوریتم تشخیص چهره را با استفاده از Imagga API پیاده سازی می

کند و امتیاز شباهت را محاسبه می‌کند.

Email Sender: عملکرد ارسال ایمیل را مدیریت می‌کند و به سیستم اجازه می‌دهد تا

کاربران را در مورد نتایج احراز هویت آگاه کند.

مراحل اجرای پروژه

ثبت نام کاربر:

کاربر اطلاعات شخصی، ایمیل و تصاویر را از طریق رابط کاربری ارائه می‌دهد.

Go داده ها را تأیید می‌کند، تصاویر را در Amazon S3 ذخیره می‌کند و اطلاعات کاربر

را در پایگاه داده PostgreSQL ثبت می‌کند. سپس یک پیام جدید را در صف RabbitMQ قرار

می‌دهد.

تشخیص چهره:

تصاویر ذخیره شده در آمازون S3 برای استخراج و تشخیص ویژگی‌های چهره بازایی می‌شوند.

اسکرپیت‌های پایتون از Imagma API برای تشخیص و تشخیص ویژگی‌های چهره استفاده می‌کنند.

سپس اطلاعات کاربر داخل پایگاه داده PostgreSQL بروزرسانی می‌شود. ایمیلی برای کاربران ارسال می‌شود و نتیجه احراز هویت را به آنها اطلاع می‌دهند.

سرویس‌های مورد استفاده:

میزبان ابری: در این پروژه از یک عدد VPS شخصی خودم در دیتاسنتر Hetzner استفاده کردم این دیتاسنتر در آلمان قرار دارد و ماشین‌های مجازی را بر بستر ابر ارائه می‌دهد.

پایگاه داده به عنوان خدمت: در این پروژه از پایگاه داده PostgreSQL ارائه شده توسط Aiven استفاده کردم. علت انتخاب این پایگاه داده در ابتدا آشنایی بیشتر خودم با آن بود و اینکه این پایگاه داده رابطه‌ای بوده و به شکل گسترده‌ای مورد استفاده قرار می‌گیرد.

S3: در ای پروژه از ذخیره سازی ابر آروان (: به عنوان ذخیره ساز تصاویر استفاده کردم. علت اصلی این انتخاب رایگان بودن و در دسترس بودن آن بود.

صف پیام: در این پروژه از RabbitMQ ارائه شده توسط cloudamqp استفاده کردم. این سرویس هم رایگان است و هم قابل دسترس. از طرفی استفاده از RabbitMQ بسیار راحت بود و به علت استفاده از پروتکل AMQP کتابخانه‌ها و پکیج‌های بسیاری از آن پشتیبانی می‌کنند.

پردازش تصویر: از IMGGA استفاده شد. این سرویس علاوه بر رایگان بودن بسیار راحت قابل استفاده است. مستندهای آن بسیار آسان هستند و در مدت کمی استفاده از آن را یاد گرفتیم. ویژگی خوب دیگر قابلیت استفاده از RESTful API برای استفاده از آن بود.

ارسال ایمیل: از MailGun برای ارسال ایمیل استفاده شد. این سرویس هم رایگان بود اما مشکل اصلی نیاز به تایید کاربران برای دریافت ایمیل بود. استفاده از سرویس گوگل می‌توانست بهتر باشد.

نتیجه گیری: پروژه Image Authenticator یک سیستم تشخیص چهره ایمن و کارآمد را با موفقیت پیاده سازی می‌کند و از احراز هویت دقیق کاربر اطمینان می‌دهد. با ترکیب عملکرد Go و تطبیق پذیری Python، این پروژه به یک گردش کاری یکپارچه، از ثبت نام کاربر تا اعلان‌های نتیجه احراز

هویت، دست می یابد. این پروژه پایه‌ای برای سیستم‌های احراز هویت مقیاس‌پذیر و امن فراهم می‌کند و تجربه و امنیت کاربر را در برنامه‌های مختلف افزایش می‌دهد.