

# طراحی سیستم اجاره‌دهی منابع ابری

## پروژه‌ی درس پایگاه داده

آذر ۹۸

هدف از این پروژه، پیاده‌سازی یک سیستم برای مدیریت یک سازمان اجاره‌دهی منابع ابری است. سناریوی استفاده از این سیستم در بخش ادامه ارائه می‌شود. شما باید با مطالعه‌ی دقیق این متن، پایگاه داده‌ی مورد نیاز برای چنین سیستمی را طراحی کرده و یک رابط کاربری ساده برای استفاده از آن طراحی کنید.

در یک سامانه‌ی توزیع منابع ابری، کاربران می‌توانند منابع مورد نیاز خود را برای عملیات پردازش، ذخیره‌سازی، میزبانی سرور و ... برای مدتی اجاره کنند. با توجه به ظرفیت‌های این سامانه‌ی توزیع، کاربران می‌توانند منابع خود را با پیکربندی‌های مختلفی ایجاد کنند. انتخاب کاربران در پیکربندی، محدود به گزینه‌های ارائه شده توسط سامانه می‌باشد. هر منبع موجود در این سامانه، سیستم‌عامل، ظرفیت RAM، ظرفیت تعداد هسته و فرکانس CPU، ظرفیت فضای ذخیره‌سازی و پهنای باند شبکه‌ی خاص خود را دارد. کاربر تنها می‌تواند منابعی را ایجاد کند که سیستم‌عامل آن در سامانه موجود باشد، و میزان RAM، CPU، فضای ذخیره‌سازی و پهنای باند آن کمتر یا مساوی با ظرفیت‌های ذکر شده برای آن منبع در سامانه باشد.

کاربران پیش از هر چیز باید در این سیستم ثبت‌نام کنند. کاربر در هنگام ثبت‌نام باید کد ملی، نام و نام خانوادگی، آدرس ایمیل و رمز عبور را وارد کند. تاریخ ثبت‌نام کاربر نیز در سیستم ثبت می‌شود. هر کاربر می‌تواند تعدادی کلید SSH داشته باشد که در هنگام ایجاد منبع، به آن اختصاص داده شود (هر کلید SSH نام و مقدار خاصی دارد). پرداخت‌ها در این سامانه از طریق کیف پول کاربران انجام می‌شود. موجودی کیف پول هر کاربر، قبل از ایجاد یک منبع باید از هزینه‌ی روزانه‌ی آن سرور بیشتر باشد، همچنین کاربران می‌توانند برای تغییر اندازه و مشخصات یک منبع، درخواست بدهند، در صورتی درخواست کاربر تایید می‌شود که موجودی کیف پول او از هزینه‌ای که جدیداً برای منبع محاسبه شده بیشتر باشد.

هر کاربر می‌تواند همزمان چندین منبع داشته باشد، برای هر منبعی که کاربر ایجاد می‌کند، علاوه بر اطلاعات پیکربندی منبع، تاریخ ایجاد، کلید احراز هویت SSH و هزینه‌ی روزانه‌ی منبع ذخیره می‌شوند. هزینه‌ی روزانه‌ی منبع، یک مقدار ثابت نیست و باید در صورت تغییر مشخصات سیستم تغییر کند. هزینه‌ی روزانه‌ی منبع با توجه به پیکربندی آن، با فرمول زیر محاسبه می‌شود:

$$(CPU\_Cores * CPU\_Frequency(GHz) * 5000) + (RAM * 4000) + (Storage * 2000) + (Bandwidth * 1000)$$

علاوه بر موارد فوق، هر کاربر می‌تواند در هر زمان که مایل بود از منبع خود Snapshot تهیه کند تا بتواند در صورت خراب شدن منبع آن را بازسازی کند؛ این Snapshot ها می‌توانند متعلق به یک منبع باشند و تنها زمان ساخت آن‌ها متفاوت باشد. این سامانه برای هر Snapshot زمان ساخت و سایز آن‌ها را ثبت می‌کند.

در این سامانه، در صورتی که کاربران دچار مشکلی بشوند، باید بتوانند مشکل خود را در قالب یک تیکت برای پشتیبانی فنی ارسال کنند. هر تیکت یک تاریخ ایجاد، پاسخ، تاریخ پاسخ‌دهی، و وضعیت دارد. وضعیت هر تیکت می‌تواند یکی از سه مقدار «منتظر پاسخ»، «پاسخ داده شده» و «لغو شده» باشد. این تیکت‌ها توسط ادمین سیستم (مدیر یا پشتیبانی فنی) پاسخ‌دهی می‌شوند.

ادمین سیستم به تمامی اطلاعات موجود در سیستم دسترسی داشته و می‌تواند آن‌ها را ویرایش یا حذف کند. همچنین مدیر می‌تواند چندین گزارش از سامانه درخواست بدهد:

- لیست کاربران با مشخصات آن‌ها
- منابع هر کاربر
- لیست منابع موجود در سامانه
- مشاهده‌ی تمامی تیکت‌ها بر اساس وضعیت

### نکات پیاده‌سازی:

- ذخیره‌سازی رمز در دیتابیس باید به صورت Hash شده‌ی ترکیب رمز و یک عبارت Salt باشد. Salt یک رشته‌ی بلند از کاراکترهای رندوم است که به عبارت رمز اضافه شده و بعد از Hash شدن به همراه عبارت Hash شده در پایگاه داده ذخیره می‌شود.
- زبان مورد استفاده برای پیاده‌سازی منطق و رابط گرافیکی این پروژه، باید Python باشد.
- دیتابیس مورد استفاده برای پروژه، PostgreSQL است.
- تا حد امکان پیاده‌سازی منطق را در سمت دیتابیس انجام دهید (با استفاده از توابع و تریگرها)

### بخش‌های اضافه بر پروژه:

- امکان جستجوی کلمه بین متن تیکت‌های سامانه با استفاده از قابلیت Full Text Search و ایندکس‌ها در PostgreSQL
- وضعیت تیکت‌ها لزوماً بعد از اولین پاسخ به «پاسخ‌داده‌شده» تغییر نمی‌کند، ممکن است ادمین سامانه از کاربر بخواهد تا اطلاعات بیشتری فراهم کند. در این صورت نیاز است تا تیکت‌هایی که در ادامه‌ی یکدیگر ایجاد می‌شوند به نحوی به یکدیگر مرتبط شوند تا بتوان مجموعه‌ی آن‌ها را پیدا کرد. به این منظور نیاز به در نظر گرفتن صفتی برای مرجع هر تیکت است. پیاده‌سازی این امکان در دیتابیس و ایجاد این قابلیت در برنامه، نمره‌ی مثبت خواهد داشت.

موفق باشید!