



درس پایگاه داد: پروژه نهایی

استاد: خانم دکتر پیشگو

دستیار ارشد: پژمان زیوری

دستیاران ناظر: محمد عذیری، مرسده رسنواد، سینا معرفت

دستیاران طراح: سینا معرفت، آیه ذوقی، آرمین اصغری، فاطمه آزادیان

## سایت خرید و فروش کالا نو و دست دوم

مقدمه:

این پروژه به دانشجویان فرصت می‌دهد تا تجربه‌ی عملی در زمینه‌ی طراحی و پیاده‌سازی پایگاه داده را کسب کنند. در این پروژه مرحله به مرحله طراحی یک وب سایت با شروع از طراحی پایگاه داده تا طراحی UI را قدم به قدم پیش خواهید برد. با استفاده از پایگاه داده‌های رابطه‌ای مانند PostgreSQL یا MySQL و ابزارهای غیر رابطه‌ای Redis برای ذخیره‌سازی اطلاعات کاربران، می‌توانید یک سایت برای خرید و فروش کالاهای دست دوم و نو ایجاد کنید. ایده این سایت خرید و فروش بی واسطه کالا و خدمات است. در این سایت، کاربران می‌توانند آگهی‌های خود را ثبت کنند، آگهی‌های دیگران را مشاهده کنند، جستجو کنند و با آگهی‌دهنده تماس بگیرند.

این پروژه دارای ۴ فاز اجباری ۱. ER ۲. Table Design & Queries ۳. Backend Side ۴. UI Side و بخش‌های امتیازی است.

### طراحی مدل‌ها:

پروژه دارای چند مدل اصلی است که وجود آن‌ها در این پروژه الزامی است و با نبود آنها ساختار پروژه دچار نقص خواهد شد و به تبع منجر به کسر نمره از دانشجو می‌شود. طبیعتاً پیاده‌سازی کامل تر پروژه منجر به دریافت نمره امتیازی بیشتر خواهد شد.

به صورت کلی می‌خواهیم Logic های زیر را داخل پروژه پیاده کنیم:

۱. کاربر ها امکان ثبت نام و ورود به سایت را دارا هستند این ورود و خروج بصورت ورود از طریق کد یکبار مصرف ارسال شده به ایمیل یا شماره تماس کاربر هستند. کاربر ها دارای دسته بندی کاربر عادی و پشتیبان وبسایت هستند.
۲. کاربر ها امکان ثبت آگهی دارند. کاربران میتوانند یک آگهی با مشخصات مدنظر خود در دسته بندی مورد نظر خود ثبت کنند و امکان شخصی سازی آگهی (متن توضیحات، قیمت، عکس ها، دسته بندی و توضیحات اضافه) را دارند.
۳. آگهی ها پس از ثبت توسط مشتریان نیازمند تایید توسط پشتیبان را دارند. پشتیبان ها امکان رد کردن آگهی به همراه دلیل رد شدن را دارند (آگهی های رد شده امکان ثبت دوباره برای بررسی پشتیبان را دارند).
۴. کاربر ها می توانند آگهی های اخیر کاربران دیگر را مشاهده کنند و جزئیات آن ها را مشاهده کنند.

۵. کاربران می‌توانند لیست آگهی‌های خود به همراه وضعیت ثبت (رد شده یا ثبت شده) و تعداد بازدید آگهی را مشاهده هستند.

۶. کاربر ها امکان جستجو توسط اسم و تیتراژ آگهی و یا جستجو از طریق یک دسته بندی خاص را دارا هستند.

۷. کاربران می‌توانند روی جستجوهای خود فیلتر اعمال کنند مانند: عکس دار بودن آگهی، کمتر یا بیشتر بودن قیمت آگهی از یک مقدار خاص و ...

۸. کاربران می‌توانند روی یک آگهی گزارش ثبت کنند گزارش شامل یک متن و یک دسته بندی گزارش آگهی است.

۹. کاربران امکان تعریف کسب و کار دارند و میتوانند مغازه خود را ثبت کنند و آگهی‌های مربوط به کسب و کار خود را ثبت کنند و از این پس آگهی‌های ثبت شده توسط کاربران از جانب مغازه ثبت شده خود قابل مشاهده خواهد بود.

۱۰. پشتیبان می‌تواند گزارش‌های ثبت شده روی آگهی‌ها را مشاهده کند و در صورت لزوم آنها را حذف کند یا کاربر را موقتاً غیرفعال کند. در این حالت کاربر وضعیت آگهی خود را حذف شده می‌بیند و دیگر کاربران امکان بازدید از آگهی را ندارند

مدل‌های مورد نیاز در این پروژه بصورت زیر است توجه کنید مدل‌های زیر مدل‌های اصلی نیستند و قابل شکسته شدن به چند مدل زیر مجموعه هستند که در طی نرمال سازی به آن خواهید رسید برای مثال آدرس میتواند دارای ترکیبی از مدل شهر و استان باشد یا وضعیت‌های آگهی مثل تعداد بازدید و غیره می‌تواند یک مدل جدا تعریف شده باشد. در کل نیازمندی‌های مساله در زیر مطرح شده است و تکمیل شدن آن بر عهده دانشجو است هدف پروژه وجود تمام لاجیک‌های توصیف شده در بالاست.

- کاربر: دارای ایمیل یا شماره تلفن، نام و نام خانوادگی، شهر فعال
- آگهی: دارای تیتراژ، کاربر ثبت کننده، وضعیت آگهی، توضیحات قیمت دسته بندی عکس (ها) محل ثبت آگهی، تاریخ ثبت و ...
- کسب و کار: نام کسب و کار، آدرس، آیدی ثبت کننده کسب و کار، دسته بندی کسب و کار و شماره ثبت
- وضعیت آگهی: آیدی آگهی، نوت پشتیبان در صورت رد شدن، تاریخ آخرین وضعیت
- گزارش‌های آگهی: آیدی آگهی، دسته بندی گزارش، متن گزارش

توضیحات تکمیلی:

- پیاده سازی لاگین کاربر و پشتیبان می‌تواند جدا از هم باشد یا با همدیگر باشد. (می‌توانید یک جدول با سطح دسترسی تعریف کنید یا دوجداول مجزا در صورت تعریف جداول جدا فرم نرمال باید رعایت شود). باید در نتیجه لاگین کاربران توکن در سمت کاربر نگه داری شود. دقت کنید که باید برای لاگین دونه کاربر تمایز قائل شوید مثل استفاده از claim و ست کردن role کاربران داخل توکن.
- آگهی‌های دارای یک سری اتریبوت مشترک هستند مانند قیمت و توضیحات و عکس و... و یک سری اتریبوت مربوط به دسته بندی مخصوص مثل کارکرد خودرو تا جای ممکن باید از تعریف ستون برای داده‌های غیر مشترک خودداری کنید در صورت تعریف نمره کسر خواهد شد (برای این بخش می‌توانید از یک جدول جدا به عنوان meta استفاده کنید که داده‌ها را بصورت ترکیبی از key value نگه داری می‌کند و ارتباط به آگهی دارد)
- آگهی‌های تعریف شده تا زمانی که به وضعیت انتشار یافته نرفته اند قابل مشاهده توسط کاربران نیست.

- در صورت غیر فعال شدن کاربران آگهی های ثبت شده کاربران نیز غیرفعال می شود.
- لیست آگهی های اخیر که در صفحه اصلی مشاهده می شود دارای ترتیب زمان انتشار هستند.
- با توجه به اینکه ثبت کننده آگهی می تواند کسب و کار یا یک کاربر ساده باشد باید Foreign key های درست تعریف شود مثل یک identifier که مشخص کند نوع ثبت کننده چیست و یک id که رکورد موردنظر را نشان بدهد.

## فاز اول : طراحی: ER Diagram و جداول پایگاه داده:

در این مرحله، باید ER Diagram مربوط به مدل های طراحی شده را ایجاد کنید. برای دید بهتر از جدول دقیق تر می توانید به بخش کوئری ها رجوع کنید. این نمودار نشان می دهد چگونگی رابطه بین اجزای مختلف پایگاه داده شما است و به عنوان یک راهنمای کلی برای ساختار دیتابیس شما عمل می کند. در این مرحله شما باید روابط و ساختار پروژه خود را تعریف کنید. پیاده سازی درست دیاگرام به شما در مراحل بعدی بسیار کمک خواهد کرد با توجه به اینکه جداول در پیاده سازی آن در صورت وجود ER صحیح به شما کمک قابل توجهی خواهد کرد. در این مرحله تمامی خواسته های لاجیکال مساله باید در دیاگرام مشخص شده باشد. همچنین پس از ترسیم دیاگرام شماتیک اولیه جداول خود را روی پایگاه داده قرار دهید.

فایل های ارسالی شما یک فایل دیاگرام بصورت فایل اصلی به همراه عکس خروجی به همراه فایل متنی کوئری های درج پایگاه داده و جداول مربوطه می باشد.

استفاده از تمامی پایگاه داده های رابطه ای مجاز می باشد (ترجیح : MYSQL , POSTGRES , ORACLE , MSSQL, SQL Server)

## فاز دوم: ایجاد فرم نرمال جداول و کوئری های اولیه:

در این مرحله، براساس طراحی ER Diagram و جداول اولیه در مرحله قبل و با حل مشکلات مرحله قبل جداول جدید دیتابیس مورد نیاز را ایجاد می کنید. **جداول باید در فرم نرمال سوم باشند (3NF)**. برای هر مدل، یک جدول در پایگاه داده ایجاد می شود که اطلاعات مربوط به آن مدل را نگه می دارد. پس از این مرحله کوئری های مساله را باید با توجه به سوالات زیر و ساختار جداول خود تهیه کنید و در قالب یک فایل sql ارسال کنید فایل sql باید دارای کوئری های ساخت دیتابیس و جداول و همچنین درج اولیه چند داده برای بررسی بهتر نتیجه های کوئری های زیر است.

### بخش Query ها :

۱. نام و نام خانوادگی کاربرانی را برگردانید که تا به حال آگهی ثبت نکرده اند.
۲. نام و نام خانوادگی تمام کاربرانی که کسب و کار دارند را بدست آورید.
۳. مجموع قیمت آگهی های هر کاربر را به ازای ماه های مختلف برگردانید.
۴. مجموع قیمت آگهی های هر کاربر را به ازای ماه های مختلف برگردانید.
۵. لیست کاربرانی که حداکثر یک آگهی در هر شهر ثبت کرده باشند را برگردانید.
۶. اطلاعات کاربری را برگردانید که جدیدترین آگهی را ثبت کرده است.

۷. شماره تلفن یا ایمیل کاربرانی که میانگین قیمت آگهی های آنها از میانگین قیمت تمام آگهی ها بیشتر باشد را بدست آورید.
۸. تعداد آگهی های ثبت شده در هر دسته بندی را به همراه نام دسته بندی و تعداد برگردانید.
۹. نام ۳ کاربر با بیشترین ثبت آگهی در یک هفته اخیر را برگردانید.
۱۰. تعداد آگهی های استان تهران را به تفکیک تعداد به ازای هر شهر مشخص کنید.
۱۱. نام شهر هایی که قدیمی ترین کاربر ثبت نام شده در سیستم در آن آگهی دارد را لیست کنید.
۱۲. نام پشتیبان های سایت را لیست کنید.
۱۳. نام کاربرانی را لیست کنید که حداقل ۲ آگهی ثبت شده دارند.
۱۴. نام کاربرانی را لیست کنید که حداکثر ۲ آگهی در دسته بندی کالای دیجیتال دارند.
۱۵. ایمیل یا شماره تلفن کاربرانی که در تمام دسته بندی ها آگهی گذاشته اند را برگردانید.
۱۶. عنوان آگهی های ثبت شده امروز را با ترتیب ساعت ثبت شدن لیست کنید.
۱۷. دومین آگهی پر بازدید را بین کل آگهی ها لیست کنید.
۱۸. نام پشتیبان با بیشترین درصد رد آگهی (نسبت تعداد آگهی های رد کرده به نسبت کل آگهی های بررسی کرده) به همراه درصد را برگردانید.
۱۹. نام خانوادگی کاربر با بیشترین تعداد آگهی رد شده را به "محمدی" تغییر دهید.
۲۰. تمام آگهی های رد شده دسته بندی لوازم خانگی کاربر محمدی را حذف کنید.
۲۱. تمام آگهی های رد شده را پاک کنید.
۲۲. تعداد بازدید آگهی هایی که دیروز توسط کاربران در استان فارس ثبت شده است را ۱۰۰ عدد افزایش دهید.
۲۳. موضوع و تعداد گزارش ها را به تفکیک موضوع برای آگهی با بیشترین تعداد گزارش را بازگردانید.

### بخش Stored procedure ها :

۱. با دریافت ایمیل یا شماره تلفن عنوان آگهی های کاربر را به ترتیب زمان ثبت لیست کنید.
۲. با دریافت ایمیل یا شماره تلفن پشتیبان نام کاربرانی که حداقل یکبار توسط این پشتیبان رد شده اند را لیست کنید.
۳. با دریافت نام شهر و دسته بندی عنوان آگهی های تایید شده از استان شهر مورد نظر را لیست کنید (اگر شهر شیراز است آگهی های استان اصفهان).
۴. عبارتی را از ورودی گرفته و آگهی هایی را که آن عبارت در تیترا، نام کاربر ثبت کننده آن آگهی و یا توضیحات آن آمده باشد را برگردانید.
۵. شماره تلفن یا ایمیل کاربر را دریافت کرده و اطلاعات کاربران همشهری با آن را برگردانید.
۶. تاریخ و تعداد را به عنوان ورودی گرفته و لیست n کاربری که از آن تاریخ به بعد بیشترین تعداد آگهی را داشتند به ترتیب نزولی برگردانید.
۷. با دریافت نام دسته بندی آگهی ها رد شده مربوط به هر دسته بندی را به ترتیب تاریخ ثبت شدن لیست کنید.
۸. با دریافت موضوع گزارش کاربران با بیشترین تعداد گزارش با موضوع روی تمام آگهی های کاربر را برگردانید.

**فاز سوم: پیاده سازی سمت سرور :**

در این مرحله، شما باید سمت سرور را پیاده‌سازی کنید تا بتواند با پایگاه داده ارتباط برقرار کند و قابلیت‌های مختلف سایت را فراهم کند. این شامل ایجاد API ها و نوشتن کدهایی است که اطلاعات مورد نیاز از پایگاه داده را دریافت، ذخیره و به کاربران برگرداند. در این مرحله، از ORM ها استفاده نمی‌شود و باید کوئری‌های SQL را به صورت مستقیم نوشت. همچنین، برای ذخیره سازی کدهای OTP و مدیریت آنها، از Redis استفاده می‌شود. این فاز بخش اصلی از پروژه است که تضمین می‌کند که سایت قادر به ارتباط با پایگاه داده است و کارایی لازم را دارد.

در این مرحله باید لاجیک های موردنیاز پروژه ها به صورت API های قابل استفاده توسط وبسایت تبدیل کنید. Api های موردنیاز بصورت زیر است :

API های مورد نیاز برای سایت خرید و فروش کالا:

#### ۱. لاگین با: OTP

این API برای احراز هویت کاربران از طریق ارسال و دریافت کد OTP استفاده می‌شود. کاربر با وارد کردن شماره تلفن خود، یک کد OTP دریافت می‌کند که باید به عنوان تأییدیه وارد شود (می‌توان برای سادگی کار کد OTP را در ریسپانس برگرداند **ارتباط با یک پروایدر اس ام اس یا ایمیل دارای نمره امتیازی می باشد**).

#### ۲. ثبت نام:

این API برای ثبت نام کاربران جدید استفاده می‌شود. کاربران باید اطلاعات موردنیاز خود را از جمله نام، ایمیل، شماره تلفن و رمز عبور را وارد کنند تا حساب کاربری برای آنها ایجاد شود.

#### ۳. ثبت آگهی:

این API به کاربران اجازه می‌دهد تا آگهی‌های خود را برای فروش کالاهای دست دوم یا نو ثبت کنند. اطلاعاتی مانند عنوان، توضیحات، قیمت و تصویر محصول باید ارسال شود.

#### ۴. تایید یا رد آگهی توسط پشتیبان:

این API برای پشتیبانی و مدیریت آگهی‌ها استفاده می‌شود. پشتیبان می‌تواند آگهی‌ها را تأیید یا رد کند و به کاربران اطلاع رسانی کند.

#### ۵. گرفتن آگهی‌های اخیر:

این API برای دریافت لیست آگهی‌های اخیر و نمایش آنها به کاربران استفاده می‌شود. آگهی‌ها بر اساس تاریخ انتشار دسته‌بندی می‌شوند.

#### ۶. گرفتن جزئیات آگهی :

پس از انتخاب یک آگهی رو لیست کاربر می‌تواند جزئیات یک آگهی را مشاهده کنید در حقیقت API های لیست تنها یک پیش نمایش از آگهی ها مثل تایتل آگهی و عکس و قیمت آن مشاهده خواهند کرد در این API جزئیات بیشتر آگهی مثل توضیحات و اطلاعات تماس بررسی می‌شود. همچنین به تعداد بازدید آگهی اضافه می‌شود.

#### ۷. آپدیت پروفایل:

این API به کاربران امکان می‌دهد تا اطلاعات پروفایل خود را مانند تصویر پروفایل، نام، شهر و اطلاعات تماس به‌روزرسانی کنند.

۸. سرچ کردن توسط دسته‌بندی یا نام و فیلتر کردن آگهی‌ها:

این API به کاربران امکان می‌دهد تا بر اساس دسته‌بندی محصولات یا نام محصولات جستجو کنند. همچنین، آگهی‌ها می‌توانند بر اساس ویژگی‌های خاصی مانند قیمت یا عکس دار بودن یا نبودن فیلتر شوند.

۹. آگهی‌های کاربر:

کاربر می‌تواند وضعیت آگهی‌های خود را مشاهده کند اعم از آخرین وضعیت و تعداد بازدید آگهی خود.

۱۱. ثبت و اضافه کردن کسب و کار:

این API برای ثبت و اضافه کردن کسب و کارهای جدید برای فروش کالاهای خود استفاده می‌شود. کسب و کارها می‌توانند اطلاعات خود را ارسال کرده و در سایت ثبت شوند.

۱۲. گزارش آگهی:

کاربران می‌توانند آگهی ثبت شده را گزارش کنند.

۱۳. بررسی آگهی:

پشتیبان می‌تواند لیست گزارش‌ها را مشاهده کند.

۱۴. حذف آگهی:

پشتیبان می‌تواند پس از بررسی آگهی در صورت نیاز آگهی‌ها را حذف کند.

۱۵. محدود کردن کاربر:

پشتیبان می‌تواند پس از بررسی آگهی در صورت نیاز کاربر را موقتاً غیر فعال کند.

موارد تکمیلی:

- هر API فقط و فقط باید یک کار منحصر به فرد انجام بدهد. و در صورت نیاز به لاجیک پیچیده تر باید API های جداگانه تعریف شود.
- برای کار با پایگاه داده استفاده از هرگونه ORM **اکیدا غیر مجاز است**. استفاده از هرگونه کتابخانه که کوئری‌ها را در سطح بالا بصورت توابع در اختیار برنامه نویس قرار دهد مجاز نیست. برای اتصال به پایگاه داده امکان استفاده از کتابخانه های اتصال به پایگاه داده مثل JDBC مجاز است اما تعامل با پایگاه داده باید بصورت Raw یا کتابخانه پیاده شده توسط دانشجو باشد.
- Redis مسئول ذخیره سازی OTP هاست باید TTL مناسب برای کلید ها تعریف شود.
- همچنین Redis مسئول یک Cache Server است به این صورت که API ها مورد تکرار مانند لیست آگهی های اخیر یا پروفایل کاربران می‌تواند پس از خوانده شدن از روی دیسک دیتابیس روی مموری Redis بصورت Marshal شده ریخته شود و دفعه های بعد

از روی ردیس خوانده شود توجه: هنگام کار با **Redis** باید به یکسان و سینک بودن دیتا ها بین دیسک و ردیس باید توجه کرد و در صورت آپدیت داده روی دیتابیس مثل تغییر پروفایل کاربر **cache** باید از روی **Redis** نیز پاک شود.

## فاز چهارم : بهبود سرچ با Elastic و پیاده سازی سمت کلاینت:

در این فاز ، پیاده سازی سمت کلاینت سایت خرید و فروش کالا به عهده دارید که می تواند به صورت **native** برای سیستم عامل های مختلف مانند اندروید یا **iOS** یا به صورت **web** باشد. این بخش از پروژه برای بهبود تجربه کاربری و افزایش دسترسی پذیری سایت انجام می شود. در این مرحله فقط یک **UI** مختصر برای استفاده از **API** ها نیاز دارید هدف این مرحله تعامل با **API** ها از طریق یک **UI** است. در صورت پیاده سازی مفصل به عنوان بخش امتیازی نیز در نظر گرفته خواهد شد.

### ۱. پیاده سازی نیتیو برای اندروید یا: iOS

در صورتی که تصمیم به پیاده سازی نیتیو برای سیستم عامل های مختلف مانند اندروید یا **iOS** گرفته شود، توسعه دهندگان باید اپلیکیشن های مخصوص به هر یک از این سیستم عامل ها را طراحی و پیاده سازی کنند. این اپلیکیشن ها باید قابلیت های مشابهی با وب سایت داشته باشند اما بر اساس زبان ها و فریمورک های مخصوص به هر سیستم عامل توسعه داده می شوند. مثل : **Swift, Flutter, Kotlin, React Native**

### ۲. پیاده سازی به عنوان وب پیج:

در صورتی که تصمیم به پیاده سازی به عنوان وب پیج گرفته شود، توسعه دهندگان باید صفحات وب را بر اساس ابزار هایی مانند **HTML** ، **CSS** و **JavaScript** طراحی کنند..

### ۳. امکانات مورد نیاز:

به منظور ایجاد یک تجربه کاربری بیشتر، این پیاده سازی باید قابلیت هایی مانند ورود و ثبت نام کاربران، مشاهده آگهی ها، ثبت آگهی های جدید، مدیریت پروفایل کاربری و جستجو و فیلتر کردن آگهی ها را فراهم کند. در این مرحله با استفاده از **API** های طراحی شده از مرحله قبل می خواهیم قابلیت های مورد نیاز را از طریق یک اینترفیس حداقلی استفاده کنیم.

## بهبود پرفورمنس سرچ با Elastic Search :

در این فاز می خواهیم یک موتور جستجو داخل برنامه پیاده کنیم . با توجه به وجود **API** سرچ در مرحله قبل که روی پایگاه داده **SQL** خود پیاده کردید امکان سرچ را برای کاربران فراهم کردید. در عمل پیاده سازی سرچ روی پایگاه داده های **SQL** بسیار کند و در لود های بالا دارای ریسپانس تایم بسیار بالایی خواهیم بود . در این مرحله می خواهیم با استفاده از پایگاه داده **elastic search** روی سرچ کردن بهبود بدهیم.

- تنها ارتباط این پایگاه داده روی API سرچ خواهد بود .
- هنگام حذف و ثبت آگهی ها دیتاهای آگهی ها علاوه بر پایگاه داده NOSQL روی Elastic Search نیز داده های مورد نیاز سرچ کردن مثل موضوع و دسته بندی و قیمت نگه داری می شود.
- سینک نگه داری داده ها روی دو پایگاه داده **بسیار مهم** است (نباید داده های دو پایگاه داده باهم تداخل داشته باشند).
- از API ها فراهم شده توسط Elastic می توان در پروژه استفاده کرد استفاده از هرگونه کتابخانه مجاز است.
- نیاز است آیدی مرتبط هر آگهی در پایگاه داده SQL ای در Elastic نیز ذخیره شود تا در صورت نیاز به جزئیات آگهی توسط API شماره 6 به توان از روی دیتابیس SQL جزئیات را برگرداند.

## فاز امتیازی :

در این مرحله فاز ۴ ام را بسط می دهیم و به یک UI قابل استفاده از کاربران تبدیل می کنیم در این مرحله می خواهیم امکانات ظاهری سایت را به حداکثر برسانیم.

- در صورت پیاده سازی بصورت وب ,ساختار و دیزاین صفحات و پیاده سازی آنها باید در حد قابل قبول باشد و در تمام دیوایس ها قابل استفاده باشد (Responsive).
- در صورت پیاده سازی Native ساختار و دیزاین صفحات و پیاده سازی آنها باید در حد قابل قبول باشد.

نمره این مرحله با توجه به تمیزی و زیبایی و قابل استفاده بودن نتیجه نهایی سنجیده خواهد شد.