

## بسمه تعالی

### معماری اثبات شده برای توسعه بک اند پروژه "واچار"

یکی از اصلی ترین چالش های پروژه dad انتخاب معماری ای است با نیازهای پروژه و قابلیت های تیم همخوانی داشته باشد. برای تعیین زبان برنامه نویسی، فریم ورک و دیتابیس، جلساتی در تیم برگزار شد و معیارهایی شامل تخصص تیم، نیازمندی های پروژه و محدودیت زمانی را ارزیابی کردیم. در نهایت، پس از بررسی دقیق مزایا و معایب بین Django و Golang، انتخاب ما برای توسعه بک اند Python Django و برای دیتابیس PostgreSQL بود.

### گزینه های فنی بررسی شده برای framework:

1. Python Django

2. Golang

### معیارهای ارزیابی

1. تخصص تیم و منحنی یادگیری: آشنایی تیم با زبان و فریم ورک.
2. نیازمندی های پروژه: همخوانی با نیازهای تجاری و محدودیت های فنی.
3. محدودیت زمانی: اهمیت سرعت تحویل پروژه.
4. پشتیبانی کامیونیتی: دسترسی به منابع و کامیونیتی در صورت بروز چالش.
5. نیازهای عملکردی: تطابق معیارهای عملکردی با نیازهای تجاری.

### تحلیل مزایا و معایب

معیار	Python Django	Golang
تخصص تیم	اکثر اعضای تیم در پایتون مهارت داشتند؛ یادگیری Django نیاز به تلاش کمی داشت.	تیم در Golang مهارت داشت، اما راه اندازی فریم ورک زمان بر بود.
سرعت توسعه	ویژگی های داخلی Django (مانند Django Admin) امکان نمونه سازی سریع را فراهم کرد.	Golang نیاز به پیاده سازی دستی حتی برای ویژگی های ساده داشت.
جامعه و اکوسیستم	پشتیبانی و منابع گسترده کامیونیتی برای Django.	کامیونیتی کوچک تر اما رو به رشد با منابع محدود برای Golang.

عملکرد	نسبت به Golang کندتر، اما برای کاربرد ما که متمرکز بر داده است کافی است. (data intensive)	سریع تر و مقیاس پذیرتر برای برنامه های پیچیده.
مقیاس پذیری آینده	مقیاس پذیری متوسط	مقیاس پذیری بالا برای کاربردهای آتی.

## تصمیم نهایی

با توجه به تحلیل انجام شده، Python Django به دلایل زیر انتخاب شد:

- سهولت استفاده: ویژگی های داخلی Django توسعه را تسریع کرده و برای محدودیت زمانی ما ایده آل بود.
- پشتیبانی از پنل های ادمین: Django Admin امکان پیاده سازی سریع پنل پشتیبانی برای تیم پروداکت و پشتیبانی را فراهم کرد.
- پشتیبانی کامیونیتی: منابع گسترده کامیونیتی به رفع چالش ها کمک می کند.
- کفایت عملکردی: اگرچه کمی کندتر از Golang بود، اما برای کاربرد ما کافی بود.

## استفاده از AI برای توسعه محصول

از آن جهت که پروژه پیچیدگی بالایی ندارد و پروژه هایی شبیه به آن در گیت هاب موجودند، استفاده از AI در توسعه این پروژه بسیار کمک دهنده خواهد بود. و حالا که تصمیم گرفته ایم که از Django استفاده کنیم، کدهای نمونه با این فریم ورک به وفور در بستر اینترنت یافت می شوند و ai به خوبی برای توسعه با این فریم ورک جواب می دهد.

## انتخاب دیتابیس sql

برای دیتابیس، PostgreSQL به دلایل زیر انتخاب شد:

- قابلیت اطمینان اثبات شده: مورد استفاده در شرکت های بزرگ و نشان دهنده استحکام.
- یکپارچگی با Django: ادغام آسان با Django ORM برای توسعه کارآمد.
- جستجوی Full-Text: قابلیت های قدرتمند جستجوی Full-Text و indexing در PostgreSQL نیازهای ما در مدیریت و بازیابی داده های پیچیده را به خوبی برآورده می کند.

## ذخیره سازی تصاویر

برای ذخیره سازی تصاویر، تصمیم گرفتیم که از دیتابیس PostgreSQL استفاده کنیم. با این حال، با توجه به این که ممکن است در آینده این تکنولوژی پاسخگوی نیاز محصول نباشد و نیاز به مهاجرت به سیستم های ذخیره سازی فایل مانند Swift داشته باشیم، تصمیم بر آن شد که جداول مربوط به ذخیره سازی تصاویر کاملاً جدا باشند تا مهاجرت آن ها در آینده به راحتی انجام شود.

## استراتژی تست

برای تضمین کیفیت کد و عملکرد سیستم، استراتژی تست ما شامل موارد زیر است:

1. **Unit Testing**: برای تست منطق‌های برنامه‌نویسی و اطمینان از عملکرد صحیح توابع و ماژول‌ها.
2. **Integration Testing**: برای تست تعامل بین اجزای مختلف سیستم و اطمینان از سازگاری آن‌ها.

فریم‌ورک Django امکانات گسترده‌ای برای انجام این تست‌ها با هزینه فنی پایین فراهم می‌کند. ابزارهای تست داخلی Django به ما کمک می‌کنند تا با کمترین تلاش، تست‌های مؤثری طراحی و اجرا کنیم.

## مدیریت کاربران و کالاهای مسدود شده

برای هندل کردن کاربران و کالاهای مسدود شده، استراتژی زیر را اتخاذ کردیم:

- کاربران مسدود شده: لیست کاربران مسدود شده در یک جدول جداگانه نگهداری می‌شود. در Middleware مربوط به API هایی که نیاز به Authorization دارند، وضعیت مسدودیت کاربران بررسی می‌شود.
- کالاهای مسدود شده: در جدول مربوط به کالاها، وضعیت مسدودیت ذخیره می‌شود و در API های مربوطه، Validation های مناسب برای بررسی این وضعیت اعمال می‌شود.

## معماری UI

روش انتخاب شده: UI flow/wireframe diagram



## لینک فیگما

## معماری فرانت اند

فریم‌ورک انتخاب شده: NextJS

دلایل انتخاب:

Developer Experience

باتوجه به زمان کوتاه برای پیاده سازی، اینکه از استکی استفاده کنیم که تجربه توسعه خوب و راحتی داشته باشد، کامیونیتی مناسبی داشته باشد (که به معنای وجود تعداد زیادی مثال و کتابخانه و ... برایشون هست) و تیممان تجربه کار با آن را داشته باشد، حائز اهمیت است.

#### Performance

این مورد شاید با توجه به کوچک بودن پروژه در اولویت اول ما نباشد ولی next.js در عین تجربه توسعه مناسب، پرفورمنس خوبی هم ارائه میدهد. React پر استفاده ترین فریمورک توسعه وب است و next، به اصطلاح (جوری که خودش ادعا میکند) در واقع یک فریمورک برای React محسوب میشود. لذا علاوه بر مزیت های React، بهبود ها و موارد آماده ای را هم به آن اضافه کرده است.

#### SEO-Friendly

با توجه به ماهیت فروشگاه اینترنتی، بهینه بودن برای موتور های جست و جو یک مزیت خیلی مهم برای آن به حساب می آید. next.js با بهره گیری از server side rendering، این خواسته را نیز برآورده میکند.

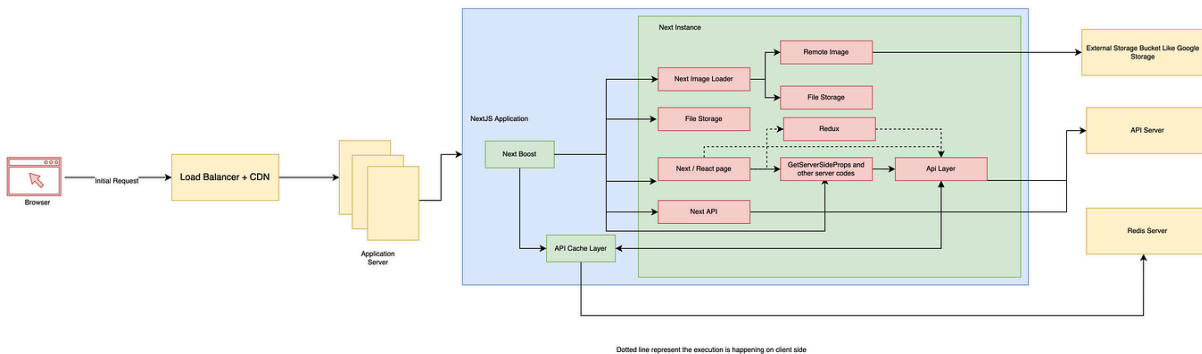
#### Knowledge

دانش استفاده از این فریمورک در تیم وجود دارد.

معماری کلی فرانت اند، با توجه به انتخاب Next.js به عنوان ستون اصلی آن تعیین میشود.

برای مثال [ساختار کلی پروژه](#) را خود Next معین میکند. (best practice)

همچنین از منابعی که در زمینه best practice های معماری یک اپلیکیشن با Next.js وجود دارد استفاده میکنیم. برای مثال این مقاله:

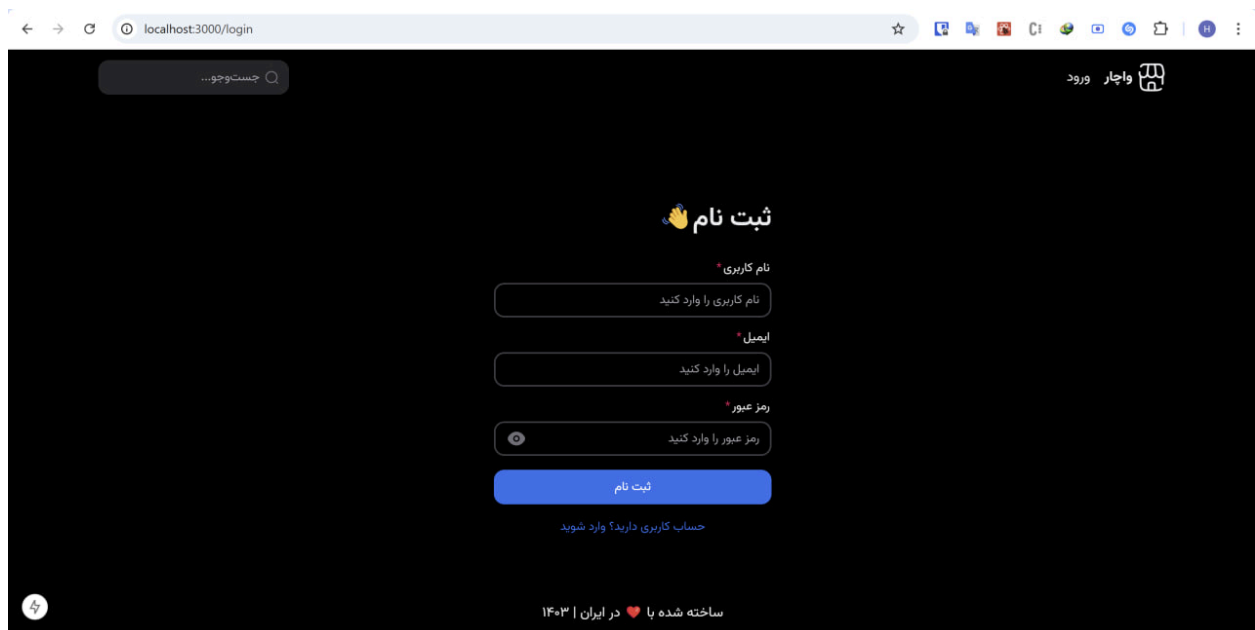


#### [Next.js Application Architecture for best performance](#)

### کتابخانه رابط کاربری: NextUI

دلایل انتخاب:

بهترین کتابخانه رابط کاربری مرتبط با Next.js بود (با توجه به میزان استفاده و star در گیتهاب)  
راه حل های آماده برای کامپوننت های پرکاربرد (مثل سرچ بار) و صفحات پرتکرار (مثل لاگین) ارائه میداد.



نمایی از صفحه ثبت نام پروژه واچار

## دیزاین اولیه دیتابیس

### - کالا

- آیدی
- استتوس معامله (فعال - رزرو - فروخته شده - مسدود شده)
- ایدی کتگوری
- قیمت
- توضیحات
- ایدی کاربری که آگهی کرده (فروشنده)
- ایدی کاربری که خریداری/رزرو
- عنوان

### - تصاویر

- ایدی
- ایدی کالا
- ترتیب
- فایل تصویر

### - کتگوری

- ایدی
- عنوان

### - یوزر

- ایدی

- نام و نام خانوادگی

- شماره تلفن

- آدرس

- ایمیل

- **تخلف**

- ایدی

- ایدی کالا

- متن

- آیدی کاربر گزارش دهنده

- استتوس (تایید شده، رد شده، در حال بررسی)

- **کاربران بلک لیستی**

- ایدی

- یوزر ایدی