به نام خدا

معماری نرم افزار جامع خرید و فروش کتاب مبتنی بر پیشنهاد هوشمند

**معماری Client-Based**

تمام پردازش‌ها و ذخیره‌سازی داده‌ها در سمت کلاینت انجام می‌شود.

سرور معمولاً برای ذخیره داده‌ها یا احراز هویت نقش ساده‌ای ایفا می‌کند.

نیاز به سرور قوی و پیچیده ندارد.

مناسب برای سیستم‌های کوچک با تعداد کاربران کم.

مدیریت و امنیت داده‌ها در کلاینت مشکل است.

امکان دسترسی چندکاربره به داده‌ها به صورت همزمان محدود است.

در صورتی که کاربران زیادی به سیستم متصل شوند، کارایی کاهش می‌یابد.

**معماری Server-Based**

تمام پردازش‌ها در سمت سرور انجام می‌شود، و کلاینت صرفاً رابطی برای نمایش اطلاعات است.

امنیت داده‌ها و کنترل آن‌ها در سمت سرور بهتر مدیریت می‌شود.

مناسب برای سیستم‌های پیچیده و متمرکز.

نیاز به سرورهای قدرتمند و مقیاس‌پذیر.

اگر سرور از دسترس خارج شود، کل سیستم از کار می‌افتد.

بار پردازشی و شبکه بالایی روی سرور اعمال می‌شود.

**معماری Client-Server-Based**

پردازش‌ها بین کلاینت و سرور تقسیم می‌شود. کلاینت بخشی از پردازش‌ها (مثل نمایش داده یا اعتبارسنجی ساده) را انجام می‌دهد و سرور مسئول ذخیره‌سازی داده‌ها و عملیات سنگین‌تر است.

تعادل در بار پردازشی بین کلاینت و سرور.

امنیت بهتر داده‌ها، زیرا بخش مهمی از پردازش‌ها در سمت سرور انجام می‌شود.

امکان مقیاس‌پذیری برای مدیریت کاربران متعدد.

مناسب برای سیستم‌های مدرن با کاربران زیاد و تعاملات پیچیده.

نیاز به طراحی دقیق برای هماهنگی بین کلاینت و سرور.

پیچیدگی بیشتر نسبت به مدل‌های ساده‌تر.

**تحلیل بر اساس نیازمندی‌های Non-Functional**

**امنیت (Security)**

* معماری Client-Server-Based امنیت داده‌ها را بهبود می‌دهد، زیرا داده‌های حساس در سرور مدیریت و ذخیره می‌شوند، نه در کلاینت. این موضوع به ویژه برای سیستم‌هایی با پرداخت آنلاین و مدیریت اطلاعات کاربران حیاتی است.

**مقیاس‌پذیری (Scalability)**

* در معماری Client-Server-Based، امکان مقیاس‌پذیری سرور برای پاسخگویی به تعداد کاربران زیاد وجود دارد. می‌توان با استفاده از Load Balancer‌ها سرورها را به صورت افقی یا عمودی مقیاس داد.

**کارایی (Performance)**

* در مقایسه با Server-Based، تقسیم بار پردازشی بین کلاینت و سرور در Client-Server-Based باعث بهبود کارایی می‌شود. به‌ویژه اگر عملیات ساده (مثل نمایش رابط کاربری یا اعتبارسنجی اولیه) در کلاینت انجام شود.

**دسترس‌پذیری (Availability)**

* در معماری Client-Server-Based، می‌توان از سرورهای پشتیبان برای افزایش دسترس‌پذیری سیستم استفاده کرد. اگر سرور اصلی از کار بیفتد، سرورهای پشتیبان جایگزین می‌شوند.

**انعطاف‌پذیری (Flexibility)**

* این معماری انعطاف بالایی دارد و می‌توان به راحتی قابلیت‌های جدید به سیستم اضافه کرد. به عنوان مثال، افزودن یک سرویس جدید (مثل سیستم پاداش) به سرور و هماهنگی با کلاینت‌ها امکان‌پذیر است.

**انتخاب مناسب معماری Client-Server-Based**

1. **تعاملات پیچیده و نیازمندی‌های امنیتی:**

سیستم شما نیازمند مدیریت تراکنش‌های مالی، نظرات کاربران، و اطلاعات حساس است. این اطلاعات باید در سمت سرور مدیریت شوند تا امنیت تضمین شود.

1. **تعداد کاربران بالا و قابلیت مقیاس‌پذیری:**

با توجه به اینکه سیستم شما ممکن است تعداد زیادی کاربر داشته باشد، نیاز به معماری‌ای دارید که بتواند به طور همزمان به تعداد زیادی کلاینت پاسخ دهد.

1. **تعادل در بار پردازشی:**

توزیع عملیات بین کلاینت و سرور باعث کاهش بار سنگین روی سرور می‌شود.

1. **تطابق با معماری‌های مدرن:**

معماری Client-Server-Based یکی از رایج‌ترین و بهینه‌ترین معماری‌ها برای سیستم‌های مدرن است، به‌ویژه اگر در آینده نیاز به ارتقا یا توسعه سیستم وجود داشته باشد.

معماری Client-Server-Based با توجه به تعادل در توزیع بار پردازشی، مقیاس‌پذیری بالا، امنیت مناسب و انعطاف‌پذیری برای توسعه‌های آتی، بهترین گزینه برای سیستم ما است.