**Software Architecture Design**

معماری مبتنی بر رویداد (Event-Driven Architecture) به عنوان الگوی مناسب برای پروژه شما انتخاب شده است.

**دلایل انتخاب معماری مبتنی بر رویداد:**

بر اساس معماری مبتنی بر رویداد برای پروژه پیشنهاد شغل بر اساس نمرات و ویژگی‌ها، دلایل زیادی وجود دارد. در زیر، ۱۰ دلیل برای انتخاب این معماری به همراه مثال‌هایی آورده شده است:

1. \*\*لیزوسینگ (Loose Coupling):\*\*

- \*مثال:\* اگر یک بخش از سیستم، مثلاً الگوریتم ارزیابی نمرات، به‌روزرسانی شود، سیستم به لحاظ معماری مبتنی بر رویداد این امکان را فراهم می‌کند که این بخش به صورت مستقل از بخش‌های دیگر تغییر یابد.

2. \*\*قابلیت مقیاس‌پذیری (Scalability):\*\*

- \*مثال:\* با توجه به این که وقایع مرتبط با نمرات و ویژگی‌ها به صورت مستقل تولید و پردازش می‌شوند، سیستم به راحتی قابلیت مقیاس‌پذیری را دارد.

3. \*\*واکنش به وقایع به صورت لحظه‌ای (Real-time Event Response):\*\*

- \*مثال:\* وقایعی مثل تغییر نمرات یا افزودن ویژگی‌های جدید به صورت فوری پردازش شده و به تولید پیشنهادها منجر می‌شوند.

4. \*\*افزایش امکانات (Enhanced Functionality):\*\*

- \*مثال:\* اگر بخواهیم به سیستم ویژگی‌های جدیدی اضافه کنیم، مثل در نظر گرفتن تجربه کاربری، معماری مبتنی بر رویداد این اضافه کردن ویژگی‌ها را آسان می‌کند.

5. \*\*مقاومت در برابر خطا (Fault Tolerance):\*\*

- \*مثال:\* در صورت بروز خطا در یکی از بخش‌ها، سایر بخش‌ها می‌توانند به صورت مستقل و به طور موازی کار خود را ادامه دهند.

6. \*\*ردیابی و نظارت بهتر (Better Tracking and Monitoring):\*\*

- \*مثال:\* هر رویداد مهم مانند تغییر نمرات به عنوان یک وقایع، به طور جداگانه قابل ردیابی است که این امکان را فراهم می‌کند تا درخواست‌ها و وضعیت پردازش‌ها بهتر نظارت شود.

7. \*\*مدیریت تراکنش‌ها (Transaction Management):\*\*

- \*مثال:\* اگر یک تراکنش مربوط به افزایش ویژگی‌ها و ارزیابی نمرات ناکام باشد، اثر آن بر سایر تراکنش‌ها به حداقل می‌رود.

8. \*\*سهولت ادغام با سیستم‌های خارجی (Ease of Integration with External Systems):\*\*

- \*مثال:\* سیستم می‌تواند به راحتی با سایر سرویس‌ها و سیستم‌های خارجی که وابستگی به این نمرات و ویژگی‌ها دارند، ادغام شود.

9. \*\*قابلیت بازیابی (Recoverability):\*\*

- \*مثال:\* در صورت بروز مشکل یا خطا، اطلاعات مربوط به وقایع ذخیره شده و به صورت مستقل قابل بازیابی است.

10. \*\*سهولت تست و توسعه (Ease of Testing and Development):\*\*

- \*مثال:\* تست و اصلاح یک بخش مستقل از سیستم (مثل محاسبه نمرات) ساده‌تر است و توسعه افزودن ویژگی‌ها نیز به‌صورت مستقل انجام می‌شود.

معماری مبتنی بر رویداد با توجه به این دلایل، برای پروژه پیشنهاد شغل با توجه به نمرات و ویژگی‌ها به عنوان یک گزینه موثر و انعطاف‌پذیر محسوب می‌شود.

**نحوه طراحی:**

مطابق با الگوی معماری مبتنی بر رویداد (Event-Driven Architecture)، یک سیستم توصیه شغل با آزمون شغل می‌تواند به شکل زیر طراحی شود:

۱. \*\*Event Producers:\*\*

- \*\*User Interaction Events:\*\* نمایانگر وقایعی مانند "Start Job Test" که از سوی کاربران ایجاد می‌شود و نشان‌دهنده شروع آزمون شغل است.

- \*\*Test Completion Events:\*\* وقایعی مانند "Job Test Completed" که توسط سیستم آزمون ایجاد می‌شود و به پایان رسیدن آزمون را نشان می‌دهد.

۲. \*\*Event Bus:\*\*

- یک Event Bus که وظیفه انتقال وقایع از تولیدکنندگان به مصرف‌کنندگان را داراست.

۳. \*\*Event Consumers - Job Recommendation System:\*\*

- یک سیستم پیشنهاد کننده شغل که به عنوان یک مصرف‌کننده رویداد عمل می‌کند. این سیستم بر اساس وقایع "Job Test Completed" الگوریتم‌های تحلیلی اجرا کرده و شغل‌های مناسب را پیشنهاد می‌دهد.

۴. \*\*Job Recommendations Events:\*\*

- وقایعی مانند "Job Recommendations Generated" که توسط سیستم پیشنهاد کننده شغل ایجاد می‌شوند و حاوی لیست شغل‌های پیشنهادی برای کاربران می‌باشند.

۵. \*\*User Interface:\*\*

- رابط کاربری که از طریق وقایع "Job Recommendations Generated" اطلاعات پیشنهاد شغل را به کاربران نمایش می‌دهد.

\*\*فرآیند عملکرد:\*\*

1. کاربر با شروع آزمون شغل یک وقایع "Start Job Test" ایجاد می‌کند که از طریق Event Bus به سیستم توصیه منتقل می‌شود.

2. سیستم آزمون آزمون را اجرا کرده و در صورت پایان موفق، یک وقایع "Job Test Completed" ایجاد می‌کند و این وقایع نیز به Event Bus ارسال می‌شوند.

3. سیستم پیشنهاد کننده شغل به عنوان یک مصرف‌کننده این وقایع را دریافت کرده و الگوریتم‌های تحلیلی را اجرا کرده و وقایع "Job Recommendations Generated" را ایجاد می‌کند.

4. این وقایع حاوی لیست شغل‌های پیشنهادی به واسطه Event Bus به رابط کاربری ارسال می‌شوند.

5. رابط کاربری اطلاعات پیشنهاد شغل را به کاربر نمایش می‌دهد.

این ساختار نمایی از چگونگی عملکرد یک سیستم توصیه شغل با آزمون شغل در یک معماری مبتنی بر رویداد را نشان می‌دهد.