فرمت انتخابی: پایگاه داده رابطه‌ای(RDBMS)

دلایل انتخاب:RDBMS

تحلیل نیازمندی‌های غیرعملکردی:(Non-Functional Requirements)

۱. **عملکرد:(Performance)**

- پروژه ما نیازمند زمان پاسخگویی کمتر از ۲ ثانیه در شرایط عادی است.
- RDBMSها با استفاده از **ایندکس‌ها (Indexes)** و **بهینه‌سازی کوئری‌ها (Query Optimization)** سرعت بازیابی داده‌ها را تضمین می‌کنند.
- همچنین، برای پردازش داده‌های تراکنشی مانند ثبت‌نام کاربران یا شارژ حساب‌ها، عملکرد مطلوبی ارائه می‌دهند.

۲. **امنیت:(Security)**

- سیستم ما باید از داده‌های حساس مانند اطلاعات کاربران، رمز عبور و جزئیات پرداخت محافظت کند.
- RDBMSها از **رمزنگاری داده‌ها(Data Encryption)** ، **کنترل دسترسی مبتنی بر نقش‌ها-Role) (Based Access Control**و **مدیریت تراکنش‌های ACID** برای تضمین امنیت داده‌ها پشتیبانی می‌کنند.

۳. **مقیاس‌پذیری:(Scalability)**

- داده‌های پروژه ما ساختاریافته هستند و به طور منظم افزایش می‌یابند. مقیاس‌پذیری عمودی (Vertical Scalability)که توسطRDBMS ها ارائه می‌شود، نیازهای فعلی ما را برآورده می‌کند.
- در صورت افزایش حجم داده‌ها، می‌توان از تکنیک‌های تقسیم‌بندی (Partitioning) یا کلاسترینگ (Clustering) برای بهبود مقیاس‌پذیری استفاده کرد.

۴. **دسترس‌پذیری:(Availability)**

- با توجه به نیاز به آپتایم ۹۹.۹٪ سیستم، مکانیزم‌های **Replication**و **Failover**در RDBMS امکان پایداری بالا را فراهم می‌کنند.
- RDBMSها همچنین از **بازیابی در برابر خطا (Failover Recovery)** برای حفظ پایداری سیستم در برابر خرابی پشتیبانی می‌کنند.

۵. **یکپارچگی داده‌ها:(Data Integrity)**

- در پروژه ما، ارتباط بین کاربران، دوره‌ها، سوالات و سازمان‌ها به شدت به **یکپارچگی داده‌ها** وابسته است.
- RDBMSبا استفاده از **کلیدهای اصلی (Primary Keys)** و **کلیدهای خارجی(Foreign Keys)** ، انسجام و صحت داده‌ها را تضمین می‌کند.

۶. **کاربرپسندی:(Usability)**

- رابط‌های SQL در RDBMS به دلیل سادگی و استاندارد بودن، کاربرپسند بوده و امکان کوئری‌های پیچیده را برای مدیران فراهم می‌کند.

تحلیل نیازمندی‌های عملکردی:(Functional Requirements)

۱. **مدیریت داده‌های ساختاریافته:**

- داده‌های پروژه ما شامل اطلاعات کاربران، دوره‌ها، آزمون‌ها و سازمان‌ها ساختاریافته هستند و نیاز به ذخیره و بازیابی منظم دارند.
- RDBMSبه دلیل توانایی بالا در مدیریت داده‌های ساختاریافته، برای این نوع داده‌ها مناسب است.

۲. **مدیریت تراکنش‌ها:(Transaction Management)**

- عملیات‌هایی مانند ثبت‌نام کاربران در دوره‌ها یا شارژ حساب‌ها باید به صورت ایمن و قابل اعتماد انجام شوند.
- RDBMSبا پشتیبانی از **ACID Transactions**تضمین می‌کند که این تراکنش‌ها به صورت کامل یا در صورت خطا به حالت اولیه بازگردانده شوند.

۳. **جستجوی پیشرفته و گزارش‌دهی:**

- پروژه ما نیازمند قابلیت جستجوی سوالات و دوره‌ها بر اساس معیارهایی مانند نام، نوع یا دسته‌بندی است.
- RDBMSابزارهای قوی برای این نوع جستجو و تولید گزارش‌های مدیریتی ارائه می‌دهد.

۴. **پشتیبانی از توسعه آینده:**

- سیستم ما قابلیت افزودن داده‌های جدید و به‌روزرسانی اطلاعات را نیاز دارد RDBMS .به دلیل استفاده از مدل‌های استاندارد، توسعه‌پذیری بالایی دارد.

چرا NoSQL انتخاب نشد؟

- داده‌های پروژه ما به شدت ساختاریافته بوده و نیاز به یکپارچگی بالا دارد، در حالی که پایگاه داده‌های NoSQL بیشتر برای داده‌های نیمه‌ساختاریافته یا غیرساختاریافته مناسب هستند.
- پروژه ما نیازمند تضمین تراکنش‌های ایمن و دقیق است، که این امر در RDBMS بهتر از NoSQL مدیریت می‌شود.
- هزینه پیچیدگی طراحی و نگهداری NoSQL برای نیازهای فعلی پروژه ما ضروری نیست.

با توجه به نیازمندی‌های پروژه، **پایگاه داده رابطه‌ای (RDBMS)** انتخاب مناسبی برای مدیریت داده‌ها در پروژه ما است. پیشنهاد می‌شود از پایگاه داده‌های **PostgreSQL**یا **MySQL**استفاده کنیم، زیرا علاوه بر امکانات پیشرفته، مقیاس‌پذیری و امنیت مطلوبی ارائه می‌دهند.

در صورت نیاز به مدیریت داده‌های نیمه‌ساختاریافته در آینده، می‌توان از یک معماری ترکیبی (Hybrid Architecture) شامل پایگاه داده‌های NoSQL برای بخش‌های خاص سیستم استفاده کرد.