



لقمه!

مقدمه

زمان خداحافظی از حافظه موقت فرارسیده است. در این پروژه، باید تمام اطلاعاتی را که در حافظه موقت نگهداری می شد، به پایگاه داده منتقل کنید تا با هر بار متوقف کردن سرویس خود، داده هایتان را از دست ندهید.

نگهداری اطلاعات در پایگاه داده

شما باید ساختار نگهداری داده هایتان در پایگاه داده را طراحی کنید و سپس تمام اطلاعات در حافظه موقت را به طور کامل به آن منتقل نمایید. بخش های زیر، گام های انجام این کار است.

• کاربران

تمام اطلاعات شخصی کاربران (شامل نام، نام خانوادگی، ایمیل و ...) و همچنین اطلاعات حساب کاربری (شامل اعتبار حساب) باید در پایگاه داده نگهداری شوند.

• رستوران ها، غذاها و مهمانی غذا

لیست غذاهایی را که از سرویس خارجی دریافت می کنید، به همراه اطلاعات آن ها باید در پایگاه داده ذخیره کنید و آن ها را به روز نگه دارید.

با هر بار دریافت رستوران ها از سرویس خارجی، لیست متفاوتی دریافت می کنید. شما باید در ابتدا هر بار بالا آمدن برنامه تان، یک بار لیست رستوران ها را دریافت کنید و اگر آن رستوران ها در پایگاه داده تان وجود نداشت، آن ها را اضافه کنید. همچنین، شما باید در پایگاه داده خود منوی رستوران ها و تعدادی که از هر غذا موجود دارند را در پایگاه داده خود نگه دارید.

برای مهمانی غذا هر ۳۰ دقیقه از سرویس خارجی غذاها را دریافت کنید و اطلاعات غذاهای جدید را در پایگاه داده قرار دهید. همچنین، درخواست های خرید از مهمانی غذا را در سطح پایگاه داده بررسی کنید و تعداد غذاها را بروزرسانی کنید.

در زمان پیاده سازی، برای راحتی کار می توانید این زمان را به یک مقدار کمتر تغییر دهید.

● سفارش‌ها

سفارش‌هایی که توسط کاربر تایید شده‌اند، باید به همراه وضعیت و اطلاعات دقیقشان در پایگاه داده ذخیره شوند تا علاوه بر رسیدگی به سفارش‌های فعلی، تاریخچه‌ای از سفارشات نیز در پایگاه داده موجود باشد.

جستجوها

در صفحه‌ی اصلی دو قسمت برای جستجو وجود دارد: یکی در میان رستوران‌ها و یکی در میان غذاها. در این فاز شما باید api مربوط به هرکدام از این جستجوها را پیاده‌سازی کنید و سپس با صداکردن آن api‌ها در سمت رابط کاربری، جستجوها را انجام دهید. دقت کنید که نیازی به یک صفحه‌ی جدید برای نمایش نتایج جستجو نیست و در همان صفحه‌ی اصلی، لیست رستوران‌ها یا رستوران‌هایی که آن غذاها را تهیه می‌کنند، نشان دهید. جستجو با استفاده از نام رستوران و یا نام غذاها انجام می‌شود. دقت کنید فرایند جستجو باید در سطح پایگاه داده صورت بگیرد.

امتیازی

هنگامی که تعداد یک نوع داده زیاد باشد، برای فرستادن لیستی از تمام آن داده در یک درخواست، حجم بالایی باید بارگذاری/بارگیری شود و در سمت کاربر نیز نمایش آن لیست زمان‌بر خواهد بود. برای جلوگیری از بروز این مشکل از صفحه‌گذاری¹ استفاده می‌شود. اگر بتوانید دریافت لیست رستوران‌ها در صفحه‌ی اصلی را با صفحه‌گذاری انجام دهید، به شما نمره‌ی امتیازی تعلق می‌گیرد. برای این کار لازم است که شماره‌ی صفحه و تعداد آیتم‌های هر صفحه به شکل پارامتر از بخش سمت کاربر به سرور ارسال شود. آیتم‌های هر صفحه‌ی درخواستی، حتما باید با کوئری دیتابیس به دست بیایند و دریافت لیست کامل از دیتابیس و پردازش روی آنها در حافظه نمره‌ای نخواهد داشت. نحوه‌ی نمایش اطلاعات صفحه‌گذاری شده در سمت کاربر می‌تواند شکل‌های مختلفی داشته باشد؛ مانند نمایش بر اساس شماره‌ی صفحه یا پیمایش بی پایان². اما برای سادگی در پیاده‌سازی این بخش در سمت کاربر، کافی است شما از یک دکمه‌ی “نمایش بیشتر” در انتهای لیست رستوران‌ها استفاده کنید. مثالی از این پیاده‌سازی را می‌توانید از [این لینک](#) ببینید.

نکات تکمیلی

- از پایگاه داده‌ی MySQL و درایور JDBC استفاده کنید.
- برای برقراری connection با پایگاه داده، شما نیاز به ابزاری برای مدیریت connection‌ها دارید. برای این کار باید از Resource Pool³ استفاده کنید. نمونه‌هایی از پیاده‌سازی و استفاده از آن را می‌توانید در [این آموزش](#) مشاهده کنید.

¹ Pagination

² Infinite scrolling

³ <https://martinfowler.com/bliki/ResourcePool.html>

● فرایند جستجو و همچنین قسمت امتیازی باید در سطح پایگاه داده صورت بگیرد؛ یعنی شما نمیتوانید تمام کاربران یا پروژه‌ها را در حافظه بارگذاری کرده و سپس فیلتر مورد نظر خود را اعمال نمایید.

● اجازه‌ی استفاده از ORM‌های موجود، از جمله **Hibernate** را ندارید. نگاشت⁴ داده‌های موجود در پایگاه داده و object‌های دامنه برعهده‌ی خود شماست.

● معماری پروژه بخش زیادی از نمره‌ی شما را تشکیل می‌دهد؛ لذا به جداسازی منطق برنامه از لایه‌ی دسترسی داده توجه کنید. به زبان دیگر، لایه‌ی منطق برنامه‌ی شما تنها با object‌های دامنه سروکار دارد و کوئری‌های پایگاه‌داده در لایه‌ی دسترسی قرار دارند. برای جداسازی منطق برنامه از پایگاه‌داده، می‌توانید از الگوی **DataMapper** استفاده کنید. مثال‌هایی از این الگو در کتاب **Patterns of Enterprise Application** وجود دارد. همچنین، نمونه‌کدی به عنوان مثال این الگو در کنار پروژه برای شما قرار داده شده‌است.

نکات پایانی

● کافی است که یکی از اعضای گروه در دو خط آدرس مخزن‌های پروژه (back-end, front-end) را در گیت‌لب و در خط‌های بعدی **Hash** مربوط به آخرین کامیت‌های این مخزن‌ها را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت‌ها مورد ارزیابی قرار می‌گیرد.

● ساختار صحیح و تمیزی کد برنامه بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.

● هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، ۵۰٪ نمره‌ی کل پروژه‌ها از گروه متقلب و تقلب‌دهنده کسر خواهد شد و در صورت تکرار، این روند ادامه خواهد داشت.

● سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

● ایمیل طراحان پروژه:

● iamirranjbar@gmail.com

● o.a.pom.h@gmail.com