

Angular Ödevi

Hedef

Basit bir blog yönetim sistemi oluşturup verilerimizi eklemek, listelemek, düzenlemek ve silmek.

Konu

SQL ödevimizde hazırladığımız veritabanına uygun olarak şimdi de bu verileri yönetebileceğimiz bir yönetim panelini Angular kullanarak tasarlamak istiyoruz. Modellerimiz şu şekilde olacaktır.

User	Post	Category	Comment
userId username email creationDate isActive	postId userId categoryId title content viewCount creationDate isPublished	categoryId name creationDate	commentId postId userId comment creationDate isConfirmed

Uygulamanın Yapısı ve Sayfalar

1. Uygulamamızda kullanıcı listesi, gönderi listesi, kategori listesi ve yorum listesi için olmak üzere 4 listeleme sayfası olmalıdır.
2. Kullanıcı listesi sayfasından yeni kullanıcı ekleme sayfasına yönlendirme olmalıdır, var olan kullanıcı düzenlenebilmelidir veya silinebilmelidir.
3. Bir kullanıcının silinebilmesi için o kullanıcıya ait gönderi veya yorum olmamalıdır. Aynı zamanda uygulamada tek kullanıcı kaldığında o kullanıcı silinememelidir. Uyarı verilebilir.
4. Gönderi listesinde gönderi içeriği gösterilmemelidir. Gönderi içeriği detay sayfasında gösterilmelidir. Gönderi detay sayfasında aynı zamanda gönderinin kullanıcı bilgisi, kategori bilgisi ve varsa gönderiye ait yorumlar da listelenmelidir.
5. Gönderi listesinde userId, postId ve categoryId ye göre üst kısma filtre eklenmek istenirse eklenebilir, artı puan kazandırır. Filtre eklenirse routing yapısı kullanılarak query params kullanımı ile filtreleri adresin sonuna ekleyebilirsiniz.
(?userId=1&postId=2&categoryId=3 gibi)
6. Gönderiler de düzenlenebilir ve silinebilir olmalıdır. Aynı zamanda yeni gönderi eklenebilir olmalıdır. Gönderinin kullanıcı bilgisi ve kategori bilgisi select kullanılarak

alınabilir. Select ile ilgili sorun yaşarsanız input ile id bilgisini direkt sayı olarak kullanıcının girmesini isteyebilirsiniz.

7. Bir gönderiye ait yorum varsa o gönderi silinememelidir, Uyarı verilmelidir.
8. Kategori listesinde kategoriler listelenmelidir, kategori detay sayfasında kategoriye ait gönderi sayısı gösterilebilir. Eğer 5. seçenekteki filtrelemeler yapıldıysa Kategori detay sayfasında seçili kategoriye ait post listesinin açılmasını sağlayan router link eklenirse bonus puan kazandırır.
9. Yeni kategori eklenebilir veya var olan kategori düzenlenebilir.
10. Yorum listesinde yorumlar listelenir. postId'ye göre filtreleme eklenirse ekstra puan kazandırır. Detay sayfası, yeni yorum ekleme sayfası ve silme özelliği olmalıdır.
11. Bu 4 sayfa 4 servis üzerinden yönetilmelidir. **UserService, PostService, CategoryService** ve **CommentService**.
12. Servislerde bizim derste işlediğimiz gibi mock (örnek) Javascript nesne dizisi şeklinde listemiz tanımlanabilir veya bir backend servisi kurma tecrübeniz önceden var ise verileri back end üzerinden **HttpClientModule** kullanarak çekebilirsiniz. (Zorunlu değil). Mock data şeklinde göstermeniz de bizim için yeterlidir.
13. Tüm listeleme sayfalarında sayfalama olmalıdır.
14. Sayfaların görünümü çok önemli olmamasına rağmen stillendirme konusunda başarılı olmayı hedeflemenizi öneriyoruz.
15. Birden çok sayfada kullanacak olduğunuz bileşenleri (component) bir defa tanımlayıp tüm sayfalarda Input ve Output kullanarak çeşitlendirmeniz faydalı olacaktır. (Örneğin; tüm sayfalarda kullanacağınız Table adında bir component yapmak gibi - listelenecek veriyi onu kapsayan sayfa componentinden (UserListComponent gibi) input olarak almayı düşünebilirsiniz.)

Modeldeki Veriler

1. SQL ödevinde <https://www.mockaroo.com/> sitesinden oluşturduğunuz verileri bu defa JSON formatında çıktısını alarak <https://www.convertsimple.com/convert-json-to-javascript/> adresinden JS nesnesine dönüştürerek kullanabilirsiniz.

Önem Verdiğimiz Kriterler

1. Bileşenlerin tekrar kullanılabilirliği
2. Input, Output kullanımı
3. Servislerin oluşturulması ve component içinde inject edilmesi
4. HTML template kodu içerisinde *ngIf, *ngFor, [(ngModel)] vb. direktiflerin kullanılmış olması
5. Sayfada istenen verileri gösterirken tercih edilebilecek Pipe kullanımları
6. Sayfada istenen verileri elde etmek için kullanılacak Javascript fonksiyonları (map, filter, find vb.)
7. **Sayfa yeniden yüklendiğinde sonradan eklenen, düzenlenen veya silinen verilerin yok olması önemli değildir. Yalnızca ilk yüklemekten sonra istenen verileri gerçekleştirebilmek önemlidir.**

Kolay gelsin.