

# İstanbul Java101 Bootcamp Bitirme Projesi

## Proje Açıklaması:

Okuduğunuz kitapları arşivlemek için bir ihtiyacınız olduğunu fark ettiniz diyelim. Kitap mağazalarında gezerken acaba ben bu kitabı almış mıydım veya okumuş muydum diye düşünmek istemiyorsunuz. En nihayetinde okuduğunuz kitabı basit bir ekrandan kaydedip okunmuşlar listesine almak istiyorsunuz. Ardından bir arama çubuğu ile kitap ismini yazıp geçmiş kayıtlarda arama yapmak istiyorsunuz. Böylece, okuduklarınız için bir arşiv oluşturmak istiyorsunuz. Bu projeyi Java101 Bootcamp'te gördüğümüz teknolojiler ile yapmaya çalışacağız. Ödev bireyseldir. O nedenle aynı ödevlere geçerli sayılmayacaktır. Arkadaşlarınızdan yardım pek tabii ki alabilirsiniz. 😊

## Proje Teknik Detayları:

### Okumuş olduğunuz bir kitabın bilgilerini kaydetme ekranı

- İş Akışı:** Okuduğunuz kitabın başlığı, basım yılı, açıklaması, yazarı veya yazarları ve kitaba dair eklemek istediğiniz not ile birlikte kaydedilecektir. Unutmayalım ki bir kitabın birden fazla yazarı olabilir. Kullanıcı ekrandan yazarı açılır bir menüden (selectbox) seçip diğer bilgileri yazı kutusu (textbox) içine yazıp girebilir. Kaydet tuşuna bastığında okuduğu kitap bilgisi veri tabanına yazılacaktır.
- Teknik Açıklama:** Kitap verilerinin girileceği ekran herhangi bir arayüz teknolojisi olabilir. Örneğin ön yüz geliştirme konusunda deneyimli olanlar herhangi bir ön yüz teknolojisini alabilir. Yahut mobil geliştirme ile ilgilenenler bir mobil ekran hazırlayabilir. Yahut Java Server Page, Java Server Face veya Spring Thymeleaf gibi server side bir ön yüz teknolojisi de kullanabilirsiniz. Projeler Maven projesi olacak şekilde Spring Boot projesi olacaktır. Ekrandan girilen veriler bir REST servisi vasıtasıyla kaydedilecektir. Spring MVC kütüphanesi bunun için tercih edilebilir. REST servisinin hangi HTTP metodunu alacağına, servisin endpoint isimlendirme kurallarına dikkat etmenizi bekliyorum. REST servisinin alt katmanlarında ise iş mantığını kurguladığınız Service Katmanı olacaktır. Service katmanı Transactional bir yapıda olacaktır. Service Katmanının bir altında ise Repository (DAO) katmanı yer alacaktır. DAO katmanı veri tabanı ile etkileşim sağlayan katmandır. Burada Spring Data kütüphanesi kullanılabilir. Ayrıca Hibernate teknolojisi kullanılacaktır. Veri tabanına atılan sorgular HQL (Hibernate Query Language) şeklinde olmalıdır. Kitapların kaydedildiği veri tabanı ilişkisel veri tabanı yönetim sistemi olacaktır. Örneğin: MySQL, PostgreSQL, Oracle veya MS-SQL Server gibi, dilediğinizi seçebilirsiniz. Kitap kaydetme akışı için Integration Test ve Mock Test yazmanız gerekmektedir. Kitabın kaydedildiğini REST servisinden veri tabanına kadar uçtan uca Integration Test olarak yazmalısınız. Projedeki kitap kaydetmeyi sağlayan iş mantığına sahip Service Katmanını ise Mock Test olarak yazmanız gerekecektir.

## Okumuş olduğunuz kitaplarda arama yapabilme ekranı

- 1- **İş Akışı:** Kaydettiğiniz kitaplar arasında arama yapabilmeyi sağlamalısınız. Bunun için bir ekran hazırlamanız gerekmektedir. Arama yapabilmek için bir yazı kutusu olacaktır ve kullanıcı bu kutuya istediği kitabın başlığını yazarak arayacaktır. Bulduğunuz kitapları ekranda bir tablo halinde ekranda göstermeniz gerekecektir. Tabloda kitap başlığı ve yazarın ismi bulunacaktır.
- 2- **Teknik Açıklama:** Yine ekran için dilediğiniz ön yüz teknolojisini seçebilirsiniz. Ekrandan aranacak kitabın başlığı girildikten sonra yanındaki "Ara" isimli tuşa basınca girilmiş olan başlık bilgisiyle arkada bir REST servisi çağrılacaktır. Kitabı başlığıyla arama işlemini bir REST servisi sağlayacaktır. Arama yapabilmek için REST servisine ekrandan girilen başlık verisi gönderilecektir. Yine arka planda Service ve DAO katmanlarınız olacaktır. Yukarıda bahsedilen teknolojileri tercih edebilirsiniz. Arama sonucu bulunan kitaplar sadece başlık ve yazar bilgisi olacak şekilde bir liste halinde REST servis vasıtasıyla istemciye (client) döndürülecektir. Yine kitabı arama işlemini REST katmanından veri tabanına kadar olacak bir süreç şeklinde Integration Test olarak yazmanız gerekmektedir. Fakat, arama işlemini gerçekleştiren Service Katmanını ise Mock Test şeklinde tasarlamanız gerekmektedir.

## Teknik Kısıt (Mutlaka Projede Olmalı!)

Projenizdeki Service Katmanı ve DAO katmanındaki tüm fonksiyonların çalışma sürelerini ölçen bir Aspect yazmanızı bekliyorum. Bu Aspect için bir Java etiketi (Annotation) tasarlayınız. Java etiketi sadece fonksiyonların tepesinde tanımlanabilecektir. Java etiketinin kestiği fonksiyonların çalışma süresini ölçebilmek için bir Aspect tasarlayınız. Bu Aspect tanımladığınız Java etiketinin kestiği fonksiyonların çalışma sürelerini ölçecektir.

Not: SOLID ve OOP prensiplere uygun kod yazanlar bonus alacaktır 😊 Fakat, en öncelikli amaç projenin çalışabilir olmasıdır. Projenin istekleri karşılayacak şekilde çalışması en önemli yüzdeliğe sahiptir.

## Kullanılabilecek Teknolojiler ve Kavramlar

- Spring Framework (Spring Boot, Spring MVC, Spring Data, Spring Test, Spring AOP)
- Hibernate Kütüphanesi
- REST Servisleri
- Transaction Kavramı
- İlişkisel Veri Tabanı (MySQL, PostgreSQL vb...)
- Mockito Kütüphanesi
- JSP, JCF, VueJS, Angular, Android, iOS gibi ön yüz ve mobil teknolojiler (istediğinizi seçebilirsiniz)

Hepinize sonsuz başarılar 😊 Gelecek için kodluyoruz ...