



## İÇİNDEKİLER

|                                       |   |
|---------------------------------------|---|
| GİRİŞ.....                            | 3 |
| Katmanlı Mimarilerin Temelleri .....  | 3 |
| Katmanlı Mimarilerin Avantajları..... | 3 |
| Güvenlik ve İşbirliği.....            | 3 |
| Örnek Uygulama Senaryosu .....        | 4 |
| İpuçları ve İlerleme .....            | 4 |
| SONUÇ.....                            | 4 |
| KAYNAKLAR.....                        | 4 |

## 1. GİRİŞ

### Katmanlı Mimarilerin Önemi ve Rolü

Yazılım geliştirme sürecinde, karmaşıklığı yönetmek ve projeleri organize etmek zorlu bir görev olabilir. Bu karmaşıklığı azaltmanın ve yazılımı daha yönetilebilir hale getirmenin bir yolu ise katmanlı mimarilerin kullanılmasıdır. Katmanlı mimari, yazılımı farklı işlevsel katmanlara bölen ve her katmanın belirli bir sorumluluğu olan bir tasarım yaklaşımıdır.

Bu makalede, katmanlı mimarilerin temellerini ve yazılım geliştirme sürecine olan etkilerini ele alacağız. İlk olarak, katmanlı mimarilerin ne olduğunu ve nasıl çalıştığını anlamak için temel prensipleri gözden geçireceğiz. Ardından, katmanlı mimarilerin sağladığı avantajları ve yazılım projelerine olan katkılarını inceleyeceğiz. Son olarak, bir örnek uygulama senaryosu üzerinden katmanlı mimarinin nasıl uygulandığını göreceğiz ve bu yaklaşımın gerçek dünya projelerinde nasıl kullanılabileceğine dair ipuçları vereceğiz.

Katmanlı mimariler, yazılım geliştirme sürecinde önemli bir rol oynamaktadır ve bu makale, bu yaklaşımın anlaşılmasına ve uygulanmasına katkıda bulunmayı amaçlamaktadır. Şimdi, katmanlı mimarilerin temellerine bir göz atalım ve bu önemli tasarım yaklaşımını daha derinlemesine keşfetmeye başlayalım.

### 2.Katmanlı Mimarilerin Temelleri

Bu bölümde, katmanlı mimarilerin temel prensipleri, çalışma mantığı ve yapısal bileşenleri ayrıntılı olarak açıklanır. Her bir katmanın işlevi ve sorumlulukları detaylı bir şekilde incelenir. Bu, okuyucuların katmanlı mimarilerin temel kavramlarını anlamalarına yardımcı olur ve konuyu daha derinlemesine incelemelerine olanak tanır.

### 3.Katmanlı Mimarilerin Avantajları

Bu bölümde, katmanlı mimarilerin sağladığı faydalar ve avantajlar geniş kapsamlı bir şekilde ele alınır. Modülerlik, yeniden kullanılabilirlik, bakım kolaylığı, ölçeklenebilirlik, test edilebilirlik ve platform bağımsızlığı gibi konuların her biri ayrıntılı olarak incelenir.

Her avantajın projeye olan katkısı ve gerçek dünya senaryolarıyla desteklenmesi sağlanır.

#### 4.Güvenlik ve İşbirliği

Yazılım güvenliği ve ekip içi işbirliği, başarılı bir yazılım projesi için kritik öneme sahiptir. Bu bölümde, katmanlı mimarilerin güvenlik açısından nasıl avantajlar sağladığı ve ekip içi işbirliğini nasıl desteklediği incelenir. Güvenlik önlemleri, ekip çalışması ve belirlilik gibi konular detaylı olarak ele alınır ve gerçek dünya örnekleriyle desteklenir.

#### 5. Örnek Uygulama Senaryosu

Bu bölümde, bir örnek uygulama senaryosu üzerinden katmanlı mimarilerin nasıl uygulandığı gösterilir. Bir web uygulamasının tasarımı adım adım açıklanır ve her bir katmanın nasıl işlev gördüğü detaylı olarak incelenir. Örnek senaryonun gerçek dünya projelerine nasıl uyarlanabileceği ve farklı senaryolara nasıl uygulanabileceği üzerinde durulabilir.

#### 6. İpuçları ve İlerleme

Bu bölüm, katmanlı mimarilerin başarılı bir şekilde uygulanması için ipuçları ve ilerleme stratejilerini sunar. Projeye uygun katmanlı mimarilerin nasıl seçileceği, uygulama sırasında dikkat edilmesi gereken noktalar ve ilerleme takibi için yöntemler detaylı olarak ele alınır. Ayrıca, karşılaşılabilecek yaygın sorunlar ve bunların nasıl ele alınabileceği üzerinde durulur.

7. Sonuç: Sonuç bölümü, makalenin ana noktalarını özetler ve katmanlı mimarilerin önemi ve geleceği hakkında bazı düşünceleri paylaşır. Katmanlı mimarilerin yazılım geliştirme sürecine sağladığı katkılar vurgulanır ve okuyuculara bir kez daha konunun önemi hatırlatılır.

#### KAYNAKLAR

- 1.Fowler, M. (2003). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.
- 2.Evans, E. (2003). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Professional.
- 3.Microsoft Docs. (<https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>)

4.Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.

5.Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

Civan karaarslan 23481002

