

Yapay Zekâ Destekli Modern Web Uygulamaları: Mimari Yaklaşımlar ve Dağıtım Stratejileri

Mert Yağcıoğlu¹

Özet

Yapay zekanın (YZ) web geliştirmeye entegrasyonu, modern web uygulamalarının tasarımını, uygulamasını ve dağıtımını yeniden şekillendiriyor. Yalnızca izole kodlama araçlarıyla sınırlı kalmak yerine, YZ giderek kullanıcı arayüzü oluşturma, arka uç servis oluşturma, otomatik test, güvenlik analizi ve sürekli dağıtım dahil olmak üzere tüm geliştirme yaşam döngüsünü destekliyor. Buna paralel olarak, modern web sistemleri bulut platformlarını, uç ortamları ve tarayıcılar içindeki istemci tarafı yürütmeyi kapsayan dağıtılmış mimariye doğru evriliyor. Bu bölüm, YZ teknolojilerinin geliştirme aşamaları ve hesaplama katmanları boyunca nasıl konumlandırıldığını inceleyerek, YZ destekli modern web uygulamalarına kapsamlı bir mimari bakış açısı sunmaktadır. Yaşam döngüsü aşamasına, mimari katmana ve entegrasyon stratejisine dayalı olarak YZ destekli web geliştirme yaklaşımlarını sistematik olarak sınıflandırmak için kavramsal bir sınıflandırma çerçevesi tanıtılmaktadır. Bu çerçeveye dayanarak, insan-YZ iş birliğinin yinelemeli ve ölçeklenebilir geliştirme süreçlerini nasıl destekleyebileceğini gösteren uçtan uca YZ odaklı bir web geliştirme hattı önerilmektedir. Ek olarak, geleneksel web geliştirme uygulamaları ile YZ destekli yaklaşımlar arasında niteliksel bir karşılaştırma sunularak, otomasyon potansiyeli, geliştirme verimliliği, güvenlik entegrasyonu ve mimari esneklik açısından farklılıklar vurgulanmaktadır. Analiz, yapay zekanın geliştiricilerin yerini almak yerine, onları destekleyen bir katman görevi gördüğünü ve kritik tasarım kararlarında insan gözetimini korurken verimliliği artırdığını göstermektedir. Genel olarak, bu bölüm, ortaya çıkan yapay zeka tabanlı web sistemlerini ve bunların mimari etkilerini anlamak için yapılandırılmış ve teknolojiden bağımsız bir temel sunmaktadır.

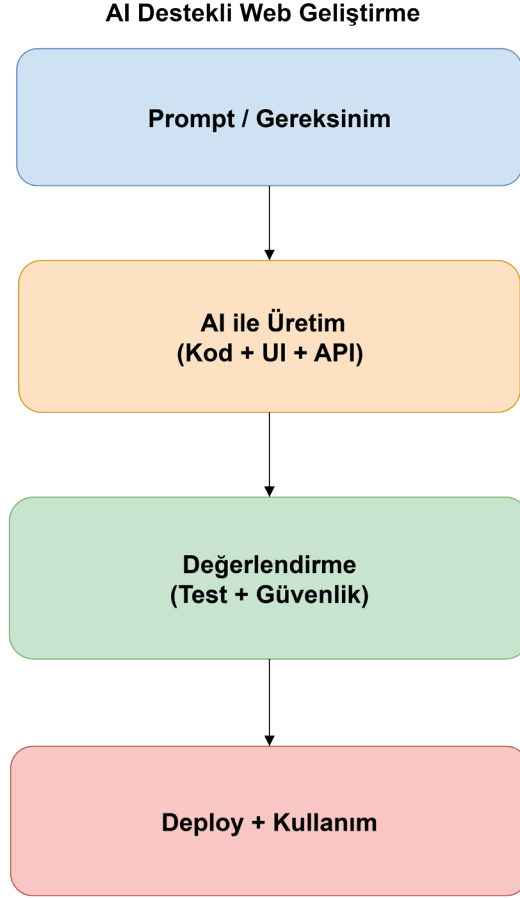
1 Dr. Öğr. Üyesi, İstanbul Arel Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, mertyagcioglu@arel.edu.tr, ORCID ID: 0000-0001-6493-6447

1. Giriş

Son yıllarda web uygulamaları, yalnızca statik içerik sunan yapılardan kullanıcı davranışlarına uyum sağlayabilen, ölçeklenebilir ve yüksek etkileşimli sistemlere dönüşmüştür. Bu dönüşüm süreci, frontend ve backend bileşenlerinin karmaşıklığını artırırken, yazılım geliştirme süreçlerinin daha kısa sürede ve daha yüksek kaliteyle tamamlanmasını zorunlu hâle getirmiştir. Bu bağlamda, yapay zekâ (YZ) tabanlı araçlar, web geliştirme süreçlerinin farklı aşamalarında giderek daha etkin bir şekilde kullanılmaya başlanmıştır.

Günümüzde yapay zekâ destekli web geliştirme araçları; kod üretimi, kullanıcı arayüzü tasarımı, hata ayıklama, otomatik test oluşturma ve güvenlik açıklarının tespiti gibi pek çok alanda geliştiricilere destek sunmaktadır. Özellikle büyük dil modelleri (Large Language Models – LLM) tabanlı sistemler, geliştiricilerin doğal dil girdileriyle fonksiyonel kod parçaları, kullanıcı arayüzü bileşenleri ve API tanımları oluşturabilmesine imkân tanımaktadır (Zou et al., 2025). Bu durum, yazılım geliştirme yaşam döngüsünde insan–makine etkileşimini yeni bir boyuta taşımaktadır.

Bununla birlikte, mevcut çalışmaların büyük bir kısmı yapay zekâ destekli araçları belirli bir geliştirme aşaması veya tekil bir teknoloji üzerinden ele almakta; bulut altyapıları, uç (edge) sistemler ve istemci tarafı (client-side) uygulamalar arasındaki bütüncül mimari entegrasyonu yeterince irdelememektedir. Oysa modern web uygulamaları, yalnızca sunucu tarafında çalışan yapay zekâ servislerinden değil, aynı zamanda tarayıcı içinde çalışan istemci tarafı modellerden ve gecikmeyi azaltmaya yönelik uç bilişim çözümlerinden de faydalanmaktadır.



řekil 1. AI destekli web geliřtirme

Bulut tabanlı yapay zekâ servisleri, yüksek hesaplama gücü ve ölçeklenebilirlik avantajları sunarken; uç biliřim ve istemci tarafı yapay zekâ yaklaşımları, düşük gecikme, veri gizliliđi ve çevrimdışı çalışma gibi kritik kazanımlar sağlamaktadır (Bulut, 2025). Bu çok katmanlı yapı, web geliřtirme süreçlerinde kullanılan yapay zekâ araçlarının yalnızca fonksiyonel deđil, aynı zamanda mimari açıdan da deđerlendirilmesini gerekli kılmaktadır. Yapay zekâ destekli web geliřtirme yaklaşımlarının, bulut, uç ve istemci katmanları arasında nasıl konumlandığını ve bu katmanlar arasındaki etkileşimi özetleyen genel görünüm řekil 1’de sunulmaktadır. řekil, yapay zekânın yalnızca sunucu tarafında çalışan bir bileřen olmadığını; tarayıcı içi modeller ve uç biliřim çözümleriyle birlikte çok katmanlı bir yapı içerisinde ele alınması gerektiğini vurgulamaktadır.

Bu çalışmada, web geliştirmede kullanılan yapay zekâ araçları ve yaklaşımları; bulut, uç ve istemci katmanlarını kapsayan bütüncül bir bakış açısıyla ele alınmaktadır. Çalışmanın temel amacı, yapay zekâ destekli web geliştirme süreçlerinde kullanılan mimarileri, frameworkleri ve entegrasyon modellerini sistematik olarak incelemek ve bu yaklaşımların modern web uygulamalarına olan katkılarını ortaya koymaktır. Ayrıca, geliştiriciler için buluttan mobil tarayıcıya kadar uzanan yapay zekâ destekli bir web geliştirme hattı (pipeline) önerilmekte ve bu hattın potansiyel avantajları tartışılmaktadır.

2. Türkiye’de Yapay Zekâ Destekli Web Geliştirme: Mevcut Eğilimler ve Odak Noktaları

Yapay zekâ (YZ) destekli araçların yazılım geliştirme süreçlerine entegrasyonu, son yıllarda hem akademiye hem de endüstride hızla artan bir ilgi alanı hâline gelmiştir. Türkiye’deki çalışmalar incelendiğinde, bu ilginin özellikle kodsuz/az kodlu (no-code/low-code) platformlar, üretken yapay zekâ ile kod üretimi, geliştirme maliyeti ve verimlilik etkileri ile YZ destekli test ve kalite değerlendirme başlıklarında yoğunlaştığı görülmektedir (Koç et al., 2025). Bu bölümün amacı, Türkiye’de yapay zekâ destekli web geliştirme alanında öne çıkan akademik eğilimleri ve çalışma odaklarını özetleyerek, çalışmanın konumlandığı bağlamı ortaya koymaktır.

2.1. Kodsuz/Az Kodlu ve Yapay Zekâ Destekli Web Geliştirme

Türkiye’de yapay zekâ destekli web geliştirme çalışmalarının önemli bir bölümü, kodsuz ve az kodlu platformlar etrafında şekillenmektedir. Bu çalışmalar, yapay zekâ ile güçlendirilmiş geliştirme ortamlarının yazılım üretim süresini kısaltma, maliyetleri düşürme ve teknik uzmanlık gereksinimini azaltma potansiyeline odaklanmaktadır. Özellikle kurumsal dijital dönüşüm süreçleri bağlamında ele alınan bu yaklaşımlar, web uygulamalarının daha hızlı hayata geçirilmesini mümkün kılan araç merkezli çözümler sunmaktadır.

Bu eksendeki çalışmalar, yapay zekânın web geliştirmede bir hızlandırıcı ve kolaylaştırıcı rol üstlenebileceğini göstermektedir (Koç et al., 2025). Ancak bu literatürde, çoğunlukla araçların sunduğu kazanımlar ön plana çıkarılmakta; bu araçların bulut, uç veya istemci katmanlarında nasıl konumlandırıldığına ya da farklı mimari bileşenlerle nasıl bütünleştiğine dair tartışmalar sınırlı kalmaktadır.

2.2. Üretken Yapay Zekâ ve Kod Üretimi Perspektifi

Üretken yapay zekâ sistemlerinin kod üretimi yetenekleri, Türkiye’deki akademik çalışmalarda da geniş yer bulmaktadır. Büyük dil modellerinin

farklı programlama dillerinde kod üretme, mevcut kodu açıklama ve hata ayıklama gibi yetkinlikleri, yazılım geliştirme süreçlerini dönüştüren önemli bir unsur olarak ele alınmaktadır (Tatlısu et al., 2025). Bu çalışmalar, genellikle üretken yapay zekâ araçlarının çıktılarının doğruluğu, kod kalitesi ve geliştirici verimliliği üzerindeki etkilerini incelemektedir.

Web geliştirme bağlamında değerlendirildiğinde, bu yaklaşımlar özellikle frontend bileşen üretimi, backend servis tanımları ve API geliştirme gibi alanlarda potansiyel faydalar sunmaktadır (Akbulut, 2025). Bununla birlikte, mevcut çalışmaların büyük bir kısmı üretken yapay zekânın genel kod üretim kabiliyetlerine odaklanmakta; web uygulamalarına özgü mimari yapıların (örneğin bileşen tabanlı frontend mimarileri veya dağıtık servis yapıları) bu araçlarla nasıl bütünleştiğini sınırlı düzeyde ele almaktadır.

2.3. İnsan–Yapay Zeka Etkileşimi ve Prompt Tabanlı Yaklaşımlar

Türkiye’de yürütülen bazı çalışmalar, yapay zekâ sistemleriyle etkileşim biçimlerini inceleyerek “prompt” kavramının çıktı kalitesi üzerindeki belirleyici rolünü vurgulamaktadır. Her ne kadar bu çalışmalar doğrudan web geliştirme odağında olmasa da elde edilen bulgular yazılım geliştirme süreçleri açısından da önemli çıkarımlar sunmaktadır. Üretken yapay zekâdan elde edilen çıktının niteliği, geliştiricinin sisteme sunduğu girdilerin açıklığına, bağlamına ve yinelemeli etkileşim döngülerine güçlü biçimde bağlıdır (Aktaş, 2025).

Web geliştirme süreçleri açısından bu durum, kullanıcı arayüzü üretimi, kod üretimi, test senaryosu oluşturma ve güvenlik analizi gibi adımların, insan denetimi altında iteratif bir biçimde yürütülmesi gerektiğini göstermektedir. Bu bağlamda, Türkiye’deki literatür insan–yapay zekâ iş birliğinin önemini dolaylı biçimde ortaya koymakta; tamamen otonom sistemlerden ziyade geliştiriciyi merkeze alan yaklaşımların öne çıktığını göstermektedir.

2.4. Test, Kalite Güvencesi ve DevOps Boyutu

Yapay zekânın yazılım test süreçlerinde kullanımı, Türkiye’deki çalışmalarda daha çok test otomasyonu ve kalite güvencesi çerçevesinde ele alınmaktadır. Bu çalışmalar, yapay zekâ tabanlı test yaklaşımlarının hata tespiti, test kapsamının genişletilmesi ve ölçeklenebilirlik açısından sunduğu avantajlara odaklanmaktadır (Ağdaş, 2025). Web geliştirme özelinde değerlendirildiğinde, bu yaklaşım özellikle üretken yapay zekâ ile üretilen kodların doğrulanması açısından kritik bir tamamlayıcı rol üstlenmektedir.

Bununla birlikte, mevcut literatürde yapay zekâ destekli test ve DevOps yaklaşımlarının, web geliştirmeye özgü uçtan uca CI/CD hatları içerisinde nasıl konumlandırılacağına dair bütüncül modeller sınırlı düzeyde ele alınmaktadır.

(Kaya, 2025). Bu durum, test, güvenlik ve dağıtım süreçlerinin yapay zekâ ile ele alındığı entegre geliştirme hatlarına yönelik ihtiyacı ortaya koymaktadır.

2.5. Genel Değerlendirme ve Çalışmanın Konumu

Türkiye'deki çalışmalar incelendiğinde, yapay zekâ destekli web geliştirme alanında ağırlıklı olarak araç odaklı, verimlilik merkezli ve belirli geliştirme aşamalarına yoğunlaşan yaklaşımların benimsendiği görülmektedir. Buna karşın, web geliştirme sürecinin tüm yaşam döngüsünü kapsayan; bulut, uç ve istemci katmanlarını birlikte ele alan bütüncül mimari çerçeveler sınırlı düzeyde ele alınmaktadır.

Bu kitap bölümünde sunulan yaklaşım, Türkiye'deki eğilimleri dikkate alarak; yapay zekâ destekli web geliştirmeyi yalnızca kullanılan araçlar üzerinden değil, mimari konumlandırma, entegrasyon biçimleri ve uçtan uca geliştirme süreçleri perspektifinden ele almayı amaçlamaktadır (Yavuz, 2025). Bu yönüyle bölüm, mevcut çalışmaları tamamlayıcı bir bağlam sunmakta ve okuyucuya alanın genel görünümünü kavramsal bir çerçeve içerisinde aktarmayı hedeflemektedir.

3. Yapay Zekâ Destekli Web Geliştirme İçin Kavramsal Sınıflandırma Çerçevesi

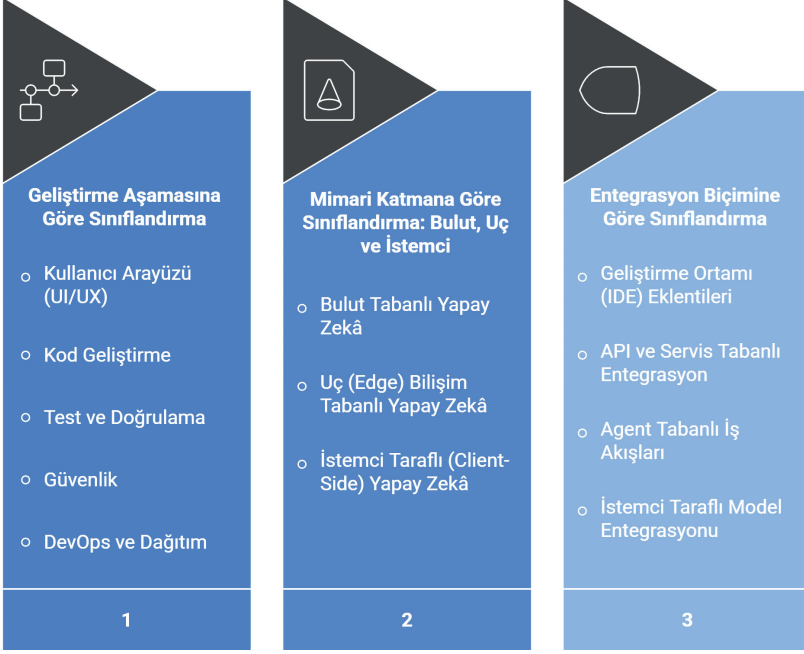
Yapay zekâ destekli web geliştirme, günümüzde çok sayıda araç, yaklaşım ve mimari çözümü kapsayan geniş ve heterojen bir alan hâline gelmiştir. Kod üretiminden kullanıcı arayüzü tasarımına, test otomasyonundan dağıtım süreçlerine kadar uzanan bu ekosistem, farklı geliştirme aşamalarında ve farklı hesaplama katmanlarında konumlanan yapay zekâ bileşenlerini içermektedir. Bu çeşitlilik, alandaki yaklaşımların yalnızca araç isimleri üzerinden ele alınmasını güçleştirmekte; daha soyut ve genellenebilir bir bakış açısını gerekli kılmaktadır.

Bu bölümde, yapay zekâ destekli web geliştirme yaklaşımlarını sistematik biçimde ele alabilmek amacıyla kavramsal bir sınıflandırma çerçevesi sunulmaktadır. Sunulan çerçeve, belirli bir araca ya da teknolojiye odaklanmak yerine, yapay zekânın web geliştirme sürecindeki rolünü, konumunu ve entegrasyon biçimini esas almaktadır. Böylece okuyucuya, hızla değişen araç ekosisteminden bağımsız olarak uzun vadede geçerliliğini koruyabilecek bir değerlendirme zemini sunulması hedeflenmektedir.

Önerilen kavramsal çerçeve üç temel boyut etrafında yapılandırılmıştır:

1. Web geliştirme yaşam döngüsündeki geliştirme aşamasına göre sınıflandırma

2. Yapay zekâ bileşenlerinin konumlandığı mimari katmana göre sınıflandırma
3. Yapay zekâ araçlarının geliştirme sürecine entegrasyon biçimine göre sınıflandırma



Şekil 2. Yapay zekâ destekli web geliştirme için sınıflandırma çerçevesi

Yapay zekâ destekli web geliştirme yaklaşımlarının sistematik biçimde ele alınabilmesi amacıyla oluşturulan kavramsal sınıflandırma çerçevesi şekil 2’de görsel olarak özetlenmektedir. Bu şekil, yapay zekânın web geliştirme süreçlerindeki rolünü; geliştirme aşaması, mimari katman ve entegrasyon biçimi olmak üzere üç tamamlayıcı boyutta ele alarak, farklı yaklaşımların ortak bir referans çerçevesi içerisinde değerlendirilmesini mümkün kılmaktadır.

Şekil 2’de sunulan bu çok boyutlu yapı, yapay zekâ destekli web geliştirme yaklaşımlarının yalnızca ne yaptığı üzerinden değil, nerede ve nasıl konumlandığı üzerinden de analiz edilmesini sağlayan bütüncül bir bakış açısı sunmaktadır.

3.1. Geliştirme Aşamasına Göre Sınıflandırma

Web uygulama geliştirme süreci, kullanıcı arayüzü tasarımından başlayarak kodlama, test, güvenlik ve dağıtım gibi birbirini izleyen ancak çoğu zaman iç içe geçen aşamalardan oluşmaktadır (Hancı & Gülseçen, 2023). Yapay zekâ

destekli araçlar, bu aşamaların her birinde farklı sorumluluklar üstlenerek geliştirme sürecinin çeşitli noktalarında değer üretmektedir.

3.1.1. Kullanıcı Arayüzü (UI/UX)

Bu aşamada yapay zekâ, doğal dil girdilerinden kullanıcı arayüzü bileşenleri üretme, taslak arayüzler oluşturma ve erişilebilirlik iyileştirmeleri önerme gibi görevlerde kullanılmaktadır (Jayaram et al., 2024). Özellikle bileşen tabanlı modern web frameworkleri bağlamında, tekrar eden arayüz yapılarını otomatik olarak oluşturabilen yapay zekâ sistemleri, geliştirme süresini önemli ölçüde azaltabilmektedir.

3.1.2. Kod Geliştirme

Kodlama aşamasında yapay zekâ; frontend ve backend bileşenlerinin üretilmesi, mevcut kodun açıklanması, yeniden düzenlenmesi ve hata ayıklanması gibi görevleri üstlenmektedir (Kalkan & Arslan, 2025). Bu bağlamda yapay zekâ, geliştiricinin yerini alan bir unsurdan ziyade, yazılım üretim sürecini hızlandıran ve destekleyen bir yardımcı olarak konumlanmaktadır.

3.1.2. Test ve Doğrulama

Yapay zekâ destekli test araçları, birim testlerinden uçtan uca test senaryolarına kadar geniş bir yelpazede otomatik test üretimi sağlayabilmektedir (Aslandođdu et al., 2022). Özellikle yapay zekâ tarafından üretilen kodların doğrulanması açısından bu aşama kritik bir tamamlayıcı rol oynamaktadır.

3.1.3. Güvenlik

Web güvenliği bağlamında yapay zekâ; kod içerisindeki potansiyel güvenlik açıklarını tespit etme, zafiyet analizi ve güvenli kodlama önerileri sunma amacıyla kullanılmaktadır (Çiçek, 2025). Bu tür araçlar, güvenliği geliştirme sürecinin erken safhalarına entegre etmeyi mümkün kılmaktadır.

3.1.4. DevOps ve Dağıtım

DevOps aşamasında yapay zekâ; CI/CD süreçlerinin otomasyonu, yapılandırma hatalarının tespiti ve dağıtım optimizasyonu gibi görevlerde kullanılmaktadır (Karagöz & Türhal, 2025). Bu durum, web uygulamalarının daha hızlı ve daha güvenilir şekilde üretim ortamına alınmasına katkı sağlamaktadır.

3.2. Mimari Katmana Göre Sınıflandırma: Bulut, Uç ve İstemci

Yapay zekâ destekli web geliştirme araçları, çalıştıkları hesaplama ortamına bağlı olarak farklı mimari özellikler sergilemektedir. Bu bağlamda araçlar; bulut, uç (edge) ve istemci (client) tabanlı yaklaşımlar altında sınıflandırılabilir (Ali & Zeebaree, 2025).

3.2.1. Bulut Tabanlı Yapay Zekâ

Bulut katmanı, yüksek hesaplama gücü ve ölçeklenebilirlik gerektiren yapay zekâ işlemleri için doğal bir ortam sunmaktadır. Büyük modellerin eğitimi ve karmaşık çıkarım işlemleri çoğunlukla bu katmanda gerçekleştirilmektedir. Ancak gecikme ve veri gizliliği gibi konular, bulut tabanlı çözümlerin dikkatle değerlendirilmesini gerektirmektedir.

3.2.2. Uç (Edge) Bilişim Tabanlı Yapay Zekâ

Uç bilişim yaklaşımı, yapay zekâ işlemlerinin kullanıcıya daha yakın noktalarda gerçekleştirilmesini sağlayarak gecikmeyi azaltmayı hedeflemektedir. Etkileşimli web uygulamaları açısından bu yaklaşım, performans ve kullanıcı deneyimi bakımından önemli avantajlar sunmaktadır.

3.2.3. İstemci Taraflı (Client-Side) Yapay Zekâ

İstemci taraflı yapay zekâ, doğrudan tarayıcı içerisinde çalışan modelleri kapsamaktadır. Veri gizliliği ve çevrimdışı çalışma gibi avantajlar sunmasına karşın, donanım kısıtları nedeniyle model karmaşıklığı sınırlı kalmaktadır. Buna rağmen, kişiselleştirme ve anlık etkileşim gerektiren senaryolarda giderek daha fazla önem kazanmaktadır.

3.3. Entegrasyon Biçimine Göre Sınıflandırma

Yapay zekâ araçlarının web geliştirme süreçlerine nasıl entegre edildiği, geliştirici deneyimini ve sistem esnekliğini doğrudan etkilemektedir. Bu bağlamda farklı entegrasyon biçimleri öne çıkmaktadır.

3.3.1. Geliştirme Ortamı (IDE) Eklentileri

Bu yaklaşımda yapay zekâ araçları, geliştiricinin kullandığı IDE veya kod editörü içerisine eklenti olarak entegre edilir. Anlık kod önerileri ve hata bildirimleri sunarak geliştirme sürecini destekler.

3.3.2. API ve Servis Tabanlı Entegrasyon

YZ yetenekleri, harici servisler aracılığıyla API üzerinden sunulur. Bu yapı, farklı web uygulamaları ve araç zincirleriyle kolay entegrasyon imkânı sağlar.

3.3.3. Agent Tabanlı İş Akışları

Agent tabanlı yaklaşımlar, birden fazla yapay zekâ bileşeninin belirli görevleri otonom veya yarı-otonom biçimde üstlendiği iş akışlarını ifade etmektedir. Bu yaklaşım, uçtan uca web geliştirme hatlarının oluşturulmasına olanak tanımaktadır.

3.3.4. İstemci Taraflı Model Entegrasyonu

Bu entegrasyon biçimi, yapay zekâ modellerinin doğrudan web uygulamasının istemci tarafına gömülmesini kapsamaktadır ve özellikle kişiselleştirme odaklı uygulamalarda tercih edilmektedir.

3.4. Genel Değerlendirme

Bu bölümde sunulan kavramsal sınıflandırma çerçevesi, yapay zekâ destekli web geliştirme yaklaşımlarını araç isimlerinden bağımsız, zamanla geçerliliğini koruyabilecek bir bakış açısıyla ele almaktadır. Geliştirme aşamaları, mimari katmanlar ve entegrasyon biçimleri birlikte değerlendirildiğinde, yapay zekânın web geliştirme süreçlerindeki rolü daha net ve sistematik bir biçimde anlaşılabilir.

Bu çerçeve, sonraki bölümlerde ele alınacak olan yapay zekâ destekli web geliştirme pipeline'ı için kavramsal bir temel oluşturmakta; okuyucuya farklı yaklaşımları karşılaştırma ve bağlamlandırma imkânı sunmaktadır.

4. Yapay Zekâ Destekli Web Geliştirme Pipeline'ı

Yapay zekâ destekli web geliştirme yaklaşımlarının pratikte değer üretebilmesi, bu yaklaşımların dağıtım araçlar ve tekil çözümler olmaktan çıkarılarak uçtan uca bir geliştirme hattı (pipeline) içerisinde ele alınmasını gerektirmektedir. Aksi takdirde, yapay zekâ yalnızca belirli aşamalarda kullanılan yardımcı bir araç olarak kalmakta; geliştirme sürecinin bütününe nüfuz edememektedir.

Bu bölümde, yapay zekânın web geliştirme yaşam döngüsünün farklı aşamalarına sistematik biçimde entegre edildiği kavramsal bir web geliştirme pipeline'ı ele alınmaktadır. Sunulan yapı, belirli bir teknolojiye ya da araca bağımlı olmaksızın; farklı ölçeklerdeki web projelerine uyarlanabilecek genel bir yol haritası sunmayı amaçlamaktadır. Pipeline, hem insan denetimini merkeze alan yarı-otonom senaryoları hem de ileri düzey otomasyon yaklaşımlarını destekleyecek şekilde kurgulanmıştır.

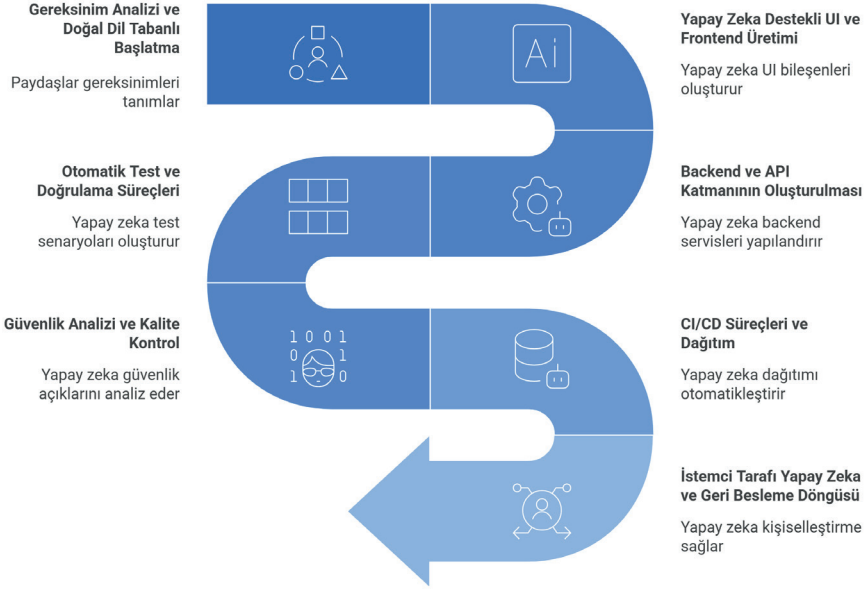
4.1. Pipeline'ın Genel Yapısı ve Tasarım İlkeleri

Yapay zekâ destekli web geliştirme pipeline'ı, klasik yazılım geliştirme süreçleriyle benzer aşamalar içerirse de bu aşamaların yapay zekâ tabanlı geri besleme döngüleri ile birbirine bağlanması bakımından farklılaşmaktadır. Pipeline, doğrusal bir akıştan ziyade, gerektiğinde önceki aşamalara geri dönülebilen iteratif ve esnek bir yapı sunmaktadır.

Bu yaklaşımın temel tasarım ilkeleri şu şekilde özetlenebilir:

- Geliştiriciyi merkeze alan insan–makine iş birliği
- Otomasyon ile denetim arasında dengeli bir yapı
- Bulut, uç ve istemci katmanlarını kapsayan dağıtık mimari uyum
- Tekrarlanabilir ve genişletilebilir geliştirme adımları

Bu ilkeler doğrultusunda ele alınan pipeline, web geliştirme sürecini yedi temel aşamada incelemektedir. Yapay zekâ destekli web geliştirme pipeline'ının temel aşamaları ve bu aşamalar arasındaki geri besleme ilişkileri Şekil 3'te gösterilmektedir. Şekil, geliştirme sürecinin doğrusal bir akıştan ziyade, yapay zekâ destekli geri besleme döngüleriyle zenginleştirilmiş iteratif bir yapı sunduğunu görsel olarak ortaya koymaktadır.



Şekil 3. Yapay zekâ destekli web geliştirme hattı

4.2. Gereksinim Analizi ve Doğal Dil Tabanlı Başlatma

Pipeline'in ilk aşaması, sistem gereksinimlerinin doğal dil kullanılarak ifade edilmesine dayanmaktadır. Bu aşamada geliştiriciler veya paydaşlar, teknik ayrıntılara girmeden uygulamanın temel işlevlerini, kullanıcı senaryolarını ve beklentilerini tanımlamaktadır. Yapay zekâ, bu girdileri analiz ederek işlevsel gereksinimler, temel bileşenler ve olası sistem mimarisi hakkında ön çıkarımlar üretmektedir.

Bu yaklaşım, teknik olmayan paydaşların da geliştirme sürecine erken aşamada katkı sağlamasına olanak tanırken; geliştirici açısından da soyut gereksinimlerin daha hızlı biçimde somut yapılarla dönüştürülmesini mümkün kılmaktadır.

4.3. Yapay Zekâ Destekli UI ve Frontend Üretimi

İkinci aşamada, tanımlanan gereksinimlere bağlı olarak kullanıcı arayüzü bileşenleri ve frontend yapıları ele alınmaktadır. Yapay zekâ, doğal dilde ifade edilen ihtiyaçlardan yola çıkarak sayfa düzenleri, arayüz bileşenleri ve temel etkileşim senaryoları üretebilmektedir.

Bu aşamada üretilen çıktılar, nihai ürün olarak değil; geliştiricinin üzerinde düzenleme yapabileceği taslak ve başlangıç noktaları olarak değerlendirilmelidir. Böylece tekrar eden arayüz tasarımlarında önemli ölçüde zaman kazanımı sağlanırken, tasarım üzerindeki insan denetimi korunmaktadır.

4.4. Backend ve API Katmanının Oluşturulması

Frontend bileşenleriyle uyumlu olacak şekilde, pipeline'in bir sonraki aşamasında backend servisleri ve API yapıları ele alınmaktadır. Yapay zekâ destekli araçlar, veri modellerinin tanımlanması, doğrulama kurallarının oluşturulması ve temel iş mantığının yapılandırılması gibi görevlerde geliştiriciye destek sunmaktadır.

Bu yaklaşım, frontend ve backend arasında sıkça karşılaşılan uyumsuzlukların azaltılmasına katkı sağlamakta; servis odaklı ve modüler mimarilerin daha hızlı biçimde hayata geçirilmesini mümkün kılmaktadır.

4.5. Otomatik Test ve Doğrulama Süreçleri

Yapay zekâ destekli web geliştirme pipeline'ında test ve doğrulama, sonradan eklenen bir adım değil; geliştirme sürecinin ayrılmaz bir parçası olarak ele alınmaktadır. Bu aşamada yapay zekâ, üretilen kod yapısına ve tanımlanan kullanıcı senaryolarına bağlı olarak birim testleri, entegrasyon testleri ve uçtan uca test senaryoları oluşturabilmektedir.

Özellikle yapay zekâ tarafından üretilen kodların güvenilirliğinin sağlanması açısından bu aşama kritik öneme sahiptir. Otomatik test üretimi, geliştiricinin test kapsamını genişletmesine yardımcı olurken, insan denetimi altında yapılan doğrulamalar sistem kalitesini desteklemektedir.

4.6. Güvenlik Analizi ve Kalite Kontrol

Pipeline'in bu aşamasında, kod ve yapılandırmalar güvenlik ve kalite açısından ele alınmaktadır. Yapay zekâ tabanlı analiz araçları, potansiyel güvenlik açıklarını, hatalı yapılandırmaları ve iyi uygulamalardan sapmaları tespit etmeye yönelik öneriler sunmaktadır.

Bu yaklaşım, güvenliğin geliştirme sürecinin geç aşamalarına bırakılmasını önleyerek, daha sürdürülebilir ve güvenli web uygulamalarının geliştirilmesine katkı sağlamaktadır.

4.7. CI/CD Süreçleri ve Dağıtım

Dağıtım aşamasında yapay zekâ, sürekli entegrasyon ve sürekli dağıtım süreçlerinin otomasyonunu desteklemektedir. Yapılandırma dosyalarının oluşturulması, dağıtım stratejilerinin seçilmesi ve kaynak kullanımının optimize edilmesi gibi görevler bu aşamada ele alınmaktadır.

Bu sayede web uygulamaları, farklı ortamlar arasında daha tutarlı ve hataya daha az açık biçimde taşınabilmektedir. Pipeline'in bu aşaması, özellikle ölçeklenebilir ve sürekli güncellenen web uygulamaları açısından kritik bir rol oynamaktadır.

4.8. İstemci Tarafı Yapay Zekâ ve Geri Besleme Döngüsü

Pipeline'in son aşamasında, istemci taraflı yapay zekâ bileşenleri web uygulamasına entegre edilmektedir. Kişiselleştirme, öneri sistemleri veya etkileşimli özellikler bu aşamada devreye alınmaktadır. Kullanıcı etkileşimlerinden elde edilen geri besleme, pipeline'in önceki aşamalarına aktarılmakta ve sistemin zaman içinde iyileştirilmesine katkı sağlamaktadır.

Özellikle istemci taraflı yapay zekâ bileşenlerinden elde edilen kullanıcı geri bildirimlerinin, pipeline'in önceki aşamalarına aktarılması, Şekil 3'te gösterildiği üzere sistemin zaman içinde kendini iyileştirebilen bir yapıya verilmesini mümkün kılmaktadır.

4.9. Bölümün Genel Değerlendirmesi

Bu bölümde ele alınan yapay zekâ destekli web geliştirme pipeline'ı, klasik yazılım geliştirme süreçlerini ikame etmeyi değil; bu süreçleri daha esnek,

daha hızlı ve daha tutarlı hâle getirmeyi amaçlayan kavramsal bir yol haritası sunmaktadır. Pipeline'ın temel gücü, yapay zekâ ile otomasyonu artırırken geliştirici denetimini koruyan dengeli yapısında yatmaktadır.

Sunulan yaklaşım, önceki bölümde tanımlanan kavramsal sınıflandırma çerçevesini somut bir geliştirme hattı ile ilişkilendirmekte ve sonraki bölümlerde ele alınacak karşılaştırmalı değerlendirmeler için doğal bir zemin oluşturmaktadır.

5. Klasik Web Geliştirme ve Yapay Zeka Destekli Web Geliştirmenin Karşılaştırmalı Değerlendirilmesi

Web uygulamalarının geliştirilme biçimi, son yıllarda yapay zekâ tabanlı araçların yaygınlaşmasıyla birlikte önemli bir dönüşüm sürecine girmiştir. Geleneksel web geliştirme yaklaşımı, uzun yıllar boyunca geliştiricinin bilgi birikimi ve manuel üretim süreçleri üzerine inşa edilmiş; tasarım, kodlama, test ve dağıtım aşamaları büyük ölçüde insan emeğiyle yürütülmüştür. Yapay zekâ destekli yaklaşımlar ise bu sürece otomasyon, öngörü ve karar destek mekanizmaları ekleyerek geliştirme pratiklerini yeniden şekillendirmektedir.

Bu bölümde, klasik web geliştirme yaklaşımı ile yapay zekâ destekli web geliştirme yaklaşımı üstünlük iddiası olmaksızın, kavramsal ve niteliksel bir bakış açısıyla karşılaştırılmaktadır. Amaç, her iki yaklaşımın güçlü ve sınırlı yönlerini ortaya koyarak okuyucuya bağlamsal bir değerlendirme sunmaktır.

5.1. Geliştirme Süreci ve Zaman Yönetimi

Klasik web geliştirme sürecinde, gereksinimlerin analiz edilmesi, kullanıcı arayüzünün tasarlanması ve kodun üretilmesi aşamaları genellikle ardışık ve manuel bir biçimde ilerlemektedir. Bu yapı, özellikle proje kapsamı genişledikçe geliştirme süresinin uzamasına neden olabilmektedir.

Yapay zekâ destekli web geliştirme yaklaşımı ise erken prototipleme ve tekrar eden görevlerin otomasyonu sayesinde geliştirme sürecini daha esnek hâle getirmektedir. Doğal dil girdilerinden taslak arayüzler veya kod bileşenleri üretilebilmesi, sürecin başlangıç aşamalarında önemli bir hızlanma sağlamaktadır. Bununla birlikte, bu hız kazanımı insan denetimi ile dengelenmediğinde, ilerleyen aşamalarda yeniden düzenleme gereksinimleri ortaya çıkabilmektedir.

5.2. Kod Üretimi ve Yeniden Kullanılabilirlik

Geleneksel yaklaşımda kod üretimi, geliştiricinin deneyimine ve bilgi birikimine doğrudan bağlıdır. Bu durum, yüksek kontrol ve özelleştirilebilirlik

sağlarken, tekrar eden kod parçalarının manuel olarak üretilmesi nedeniyle verimlilik kaybına yol açabilmektedir.

Yapay zekâ destekli yaklaşımlar, kod üretiminde yarı-otonom bir yapı sunarak tekrar eden kalıpların hızlı biçimde oluşturulmasını mümkün kılmaktadır. Bu durum, özellikle standartlaştırılmış web bileşenlerinde ve servis yapılarında yeniden kullanılabilirliği artırmaktadır. Ancak üretilen kodun kalitesinin değerlendirilmesi ve bağlamsal uygunluğunun sağlanması, geliştirici denetimini zorunlu kılmaktadır.

5.3. Test, Doğrulama ve Hata Tespiti

Klasik web geliştirme süreçlerinde test faaliyetleri çoğu zaman geliştirme sürecinin ileri aşamalarında ele alınmakta; manuel testler ve sınırlı otomasyon çözümleri yaygın olarak kullanılmaktadır. Bu yaklaşım, hataların geç tespit edilmesine ve düzeltme maliyetlerinin artmasına neden olabilmektedir.

Yapay zekâ destekli web geliştirme yaklaşımı, test üretiminin ve doğrulama süreçlerinin geliştirme hattının erken aşamalarına entegre edilmesini teşvik etmektedir. Yapay zekâ tabanlı test senaryoları, kod yapısına ve kullanıcı akışlarına bağlı olarak daha geniş bir test kapsamı sunabilmektedir. Bununla birlikte, otomatik testlerin de bağlamdan bağımsız olarak mutlak doğruluk sağladığı varsayımı, dikkatle ele alınması gereken bir konudur.

5.4. Güvenlik ve Kalite Farkındalığı

Geleneksel yaklaşımlarda web güvenliği çoğu zaman ayrı bir uzmanlık alanı olarak ele alınmakta ve geliştirme sürecinin geç safhalarında değerlendirilmekteydi. Bu durum, güvenlik açıklarının sistem mimarisine yerleşmesine neden olabilmektedir.

Yapay zekâ destekli web geliştirme, güvenlik analizlerinin geliştirme sürecine daha erken entegre edilmesine imkân tanımaktadır. Kod inceleme, zafiyet analizi ve güvenli kodlama önerileri sunan yapay zekâ araçları, geliştirici farkındalığını artırmaktadır. Ancak bu araçların sunduğu çıktılar, kesin karar mekanizmaları olarak değil; destekleyici öneriler olarak değerlendirilmelidir.

5.5. Mimari Esneklik ve Dağıtık Sistem Uyumu

Klasik web geliştirme yaklaşımları, tarihsel olarak sunucu merkezli mimariler üzerine inşa edilmiştir. Bu yapı, belirli ölçeklere kadar yeterli olmakla birlikte, modern web uygulamalarının dağıtık ve düşük gecikme gereksinimlerini her zaman karşılayamamaktadır.

Yapay zekâ destekli web geliştirme yaklaşımı, bulut, uç ve istemci katmanlarını kapsayan dağıtık mimarilere daha doğal bir uyum sağlamaktadır. Yapay zekâ bileşenlerinin farklı katmanlarda konumlandırılabilmesi, performans ve kullanıcı deneyimi açısından yeni olanaklar sunmaktadır. Bununla birlikte, bu esneklik mimari karmaşıklığı da beraberinde getirmekte ve dikkatli bir tasarım gerektirmektedir.

5.6. Geliştirici Deneyimi ve İnsan–Makine Etkileşimi

Klasik web geliştirme yaklaşımında geliştirici, sürecin merkezinde yer almakta ve tüm karar mekanizmalarını doğrudan yönetmektedir. Bu durum yüksek kontrol sağlarken, bilişsel yükü artırabilmektedir.

Yapay zekâ destekli yaklaşımlar, geliştiricinin rolünü tamamen ortadan kaldırmak yerine, karar destek sistemleri aracılığıyla yeniden tanımlamaktadır. İnsan–makine iş birliğine dayalı bu yapı, geliştiricinin yaratıcılığa ve mimari tasarıma daha fazla odaklanmasına imkân tanımaktadır. Ancak bu dönüşüm, geliştiricinin yapay zekâ çıktılarını eleştirel bir bakış açısıyla değerlendirebilme yetkinliğini de gerekli kılmaktadır.

5.7. Genel Değerlendirme

Klasik web geliştirme ve yapay zekâ destekli web geliştirme yaklaşımları, birbirini dışlayan değil; farklı bağlamlarda tamamlayıcı roller üstlenen iki paradigma olarak değerlendirilebilir. Yapay zekâ destekli yaklaşımlar, hız, otomasyon ve ölçeklenebilirlik açısından önemli avantajlar sunarken; insan denetimi, bağlamsal değerlendirme ve mimari kararlar hâlen geliştiricinin sorumluluğundadır.

Bu karşılaştırmalı değerlendirme, yapay zekâ destekli web geliştirme yaklaşımlarının mutlak bir çözüm değil; doğru bağlamda ve dengeli biçimde kullanıldığında değer üreten bir araç seti sunduğunu göstermektedir. Bir sonraki bölümde, bu yaklaşımların gelecekte nasıl evrilebileceği ve hangi araştırma alanlarının öne çıktığı ele alınacaktır.

6. Sonuç ve Gelecek Perspektifleri

Bu kitap bölümünde, yapay zekâ destekli web geliştirme alanı; kavramsal sınıflandırmalar, mimari katmanlar ve uçtan uca geliştirme süreçleri bağlamında bütüncül bir bakış açısıyla ele alınmıştır. Sunulan yaklaşım, yapay zekâyı yalnızca belirli araçlar veya kısa vadeli teknolojik eğilimler üzerinden değerlendirmek yerine, web geliştirme yaşam döngüsünün tamamına yayılan yapısal bir dönüşüm unsuru olarak konumlandırmaktadır.

Bölüm boyunca ele alınan kavramsal sınıflandırma çerçevesi, yapay zekâ destekli web geliştirme yaklaşımlarını; geliştirme aşamaları, mimari konumlandırma ve entegrasyon biçimleri açısından sistematik biçimde incelemeyi mümkün kılmıştır. Bu çerçeve, hızla değişen araç ekosistemine rağmen geçerliliğini koruyabilecek bir referans noktası sunmakta ve okuyucunun farklı yaklaşımları ortak bir dil üzerinden değerlendirmesine imkân tanımaktadır. Benzer şekilde, yapay zekâ destekli web geliştirme pipeline'ı, bu kavramsal yapıyı pratik bir yol haritası ile ilişkilendirerek, yapay zekânın geliştirme sürecine nasıl entegre edilebileceğine dair açıklayıcı bir perspektif sunmuştur.

Karşılaştırmalı değerlendirme, klasik web geliştirme ile yapay zekâ destekli yaklaşımlar arasındaki farkların mutlak üstünlükler üzerinden değil, bağlamsal uygunluk çerçevesinde ele alınması gerektiğini göstermektedir. Yapay zekâ destekli yaklaşımlar; hız, otomasyon ve ölçeklenebilirlik açısından önemli kazanımlar sağlarken, mimari kararlar, güvenlik sorumluluğu ve bağlamsal değerlendirme hâlen geliştiricinin denetiminde kalmaktadır. Bu durum, yapay zekânın web geliştirme süreçlerinde insan emeğinin yerini alan bir unsurdan ziyade, insan-makine iş birliğini güçlendiren bir destekleyici olarak ele alınmasının daha sürdürülebilir bir yaklaşım olduğunu ortaya koymaktadır.

Gelecek perspektifi açısından değerlendirildiğinde, web uygulamalarının giderek daha fazla "AI-native" bir yapıya evrileceği öngörülebilir. Yapay zekânın yalnızca geliştirme sürecinde değil, doğrudan web uygulamasının çalışma zamanında (runtime) da aktif rol üstlenmesi; kişiselleştirme, bağlamsal farkındalık ve etkileşim düzeyi yüksek uygulamaların yaygınlaşmasını beraberinde getirecektir. Bu dönüşüm, istemci tarafı yapay zekâ modelleri, uç bilişim çözümleri ve tarayıcı içi çalıştırma teknolojilerinin daha fazla önem kazanmasına yol açacaktır.

Bununla birlikte, bu evrimin beraberinde getirdiği bazı temel zorluklar da bulunmaktadır. Yapay zekâ tarafından üretilen kodun doğrulanması, güvenlik açıklarının erken tespiti, veri gizliliğinin korunması ve model bağımlılığı gibi konular, gelecekte web geliştirme alanında daha fazla tartışılacak başlıklar arasında yer alacaktır. Özellikle agent tabanlı ve yarı-otonom geliştirme iş akışlarının yaygınlaşması, etik sorumluluklar ve insan denetiminin sınırları gibi yeni araştırma alanlarını da gündeme getirmektedir.

Akademik ve endüstriyel perspektiften bakıldığında, gelecekteki çalışmaların; yapay zekâ destekli web geliştirme yaklaşımlarını gerçek dünya projeleri üzerinde uzun vadeli ve karşılaştırmalı biçimde incelemesi önem taşımaktadır. Geliştirme süresi, bakım maliyeti, güvenlik dayanıklılığı ve geliştirici deneyimi gibi ölçütlerin birlikte ele alındığı çalışmalar, bu alandaki kavramsal çerçevelerin somut etkilerini daha net biçimde ortaya koyacaktır. Ayrıca, farklı ölçeklerdeki

web uygulamaları için bulut-uç-istemci katmanlarının nasıl dengeleneceğine dair tasarım ilkeleri, önümüzdeki dönemde öne çıkan araştırma konuları arasında yer alacaktır.

Sonuç olarak, yapay zekâ destekli web geliştirme, geçici bir araç trendi olmanın ötesinde, web uygulamalarının nasıl tasarlandığını, geliştirildiğini ve sürdürüldüğünü yeniden tanımlayan yapısal bir dönüşüm sürecini temsil etmektedir. Bu kitap bölümünde sunulan kavramsal yaklaşımlar ve değerlendirmeler, okuyucuya bu dönüşümü anlamlandırma ve kendi bağlamına uyarlama konusunda rehberlik etmeyi amaçlamaktadır. Yapay zekânın olanakları ile insan uzmanlığının dengeli biçimde bir araya getirildiği yaklaşımlar, geleceğin web geliştirme pratiklerinin temelini oluşturacaktır.

Kaynakça

- Ağdaş, M. T. (2025). İnovasyonun Görünmeyen Ekibi: Yapay Zekâ Ajanları ile Akıllı Ar-Ge. *Disiplinlerarası Yaklaşımla Ar-Ge, İnovasyon ve Sürdürülebilir Kalkınma*, 167.
- Akbulut, M. (2025). Yapay Zeka Uygulamalarının Python Kod Üretimindeki Performans Karşılaştırması. *Sinop Üniversitesi Fen Bilimleri Dergisi*, 10(1), 259-288.
- Aktaş, S. G. (2025). YAPAY ZEKÂ VE GAZETECİLİK ETKİLEŞİMİNDE PROMPT MÜHENDİSLİĞİ: CHATGPT İLE KEŞİFSEL BİR ÇALIŞMA. *Gümüşhane Üniversitesi İletişim Fakültesi Elektronik Dergisi*, 13(1), 611-644.
- Ali, C. S. M., & Zeebaree, S. R. (2025). Cloud-based web applications for enterprise systems: A review of AI and marketing innovations. *Asian Journal of Research in Computer Science*, 18(4), 427-451.
- Aslandođdu, Ö. F., Aydın, H., & Çetinkaya, A. (2022). Bulut üzerinde Tensorflow Javascript ile geliştirilen Öğretilebilir Makine (ÖM) Modelinin Doğruluk Tahmini ve Analizi: Yüz Tanıma Sistemi Uygulaması. *International Journal of Management Information Systems and Computer Science*, 6(1), 66-77.
- Bulut, C. (2025). SAĞLIK KURUMLARINDA YAPAY ZEKÂ DESTEKLİ KARAR DESTEK SİSTEMLERİNİN KULLANIMI. *Uluslararası Sağlık Yönetimi ve Stratejileri Araştırma Dergisi*, 11(1), 27-37.
- Çiçek, A. E. (2025). TÜRKİYE'NİN YAPAY ZEKÂ TABANLI SİBER GÜVENLİK STRATEJİSİ: ULUSAL GÜVENLİĞİ GÜÇLENDİRMEK VE KÜRESEL SİBER YÖNETİŞİME YÖN VERMEK. *Yönetim Bilimleri Dergisi*, 23(56), 993-1012.
- HANCI, A. K., & GÜLSEÇEN, S. (2023). DevOps Test Süreç Geliştirmede Yeni Bir Model Önerisi. *Duzce University Journal of Science & Technology*, 11(4).
- Jayaram, K., Abhishek, K. A., Krishna, A., & Rayas, K. G. (2024). Rubreak: An AI-assisted UI/UX workflow implementation. In *Computer Science Engineering* (pp. 178-184). CRC Press.
- Kalkan, B., & Arslan, R. B. (2025, September). A P11Y Based Approach To Web Accessibility Evaluation. In *2025 10th International Conference on Computer Science and Engineering (UBMK)* (pp. 1281-1286). IEEE.
- KARAGÖZ, B., & TURHAL, Ü. Ç. KURUMSAL DÜZEYDE DEVOPS KÜLTÜRÜNÜN ENTEGRASYONU VE ALTYAPI OTOMASYONUNUN MODERNİZASYONU. *yaz*, 1.
- Kaya, H. (2025). Dijital Dönüşüm Çağında Yönetim Bilişim Sistemlerinin Stratejik Evrimi ve Yenilikçi Uygulamaları. *Yönetim Bilişim Sistemleri Alanında Yenilikçi Çözümler ve*, 49.

- Koç, O., Yücedağ, İ., & Şentürk, Ü. (2025). The Impact of Artificial Intelligence Enhanced No-Code Software Development Platforms on Software Processes: A Literature Review. *Duzce University Journal of Science and Technology*, 13(1), 383-401.
- Tatlisu, E., Bağlı, H. H., Turan, A. Z., Hamurcu, A., Eser, A., & Avcı, S. K. (2025). Yapay zekâ çağında endüstriyel tasarım: Uygulamalar ve eğitim üzerine etkileri. *Öneri Dergisi*, 20(MX Yaratıcı Endüstriler Çalıştayı 2024: Yapay Zeka Çağında Yaratıcı Endüstriler Özel Sayısı), 154-188.
- Yavuz, U. (2025). Üretken Yapay Zekâ Teknolojilerinin Yazılım Geliştirici İstihdamı Üzerindeki Etkileri. *Niksar Akademik Çalışmalar Dergisi*, 2(1), 26-48.
- Zou, G., Li, P., Yang, S., Hu, S., Pang, S., & Gan, Y. (2025). LLM-enhanced service Semantic Representation and Category co-occurrence feature Augmentation for Web API recommendation. *Information Processing & Management*, 62(6), 104219.