

Dönem Sonu Proje Örnekleri

HTML, CSS, JavaScript, PHP ve MySQL ile 4 İşlevsel Proje Seçeneği

Proje	Kısa açıklama	Ana beceriler
1. Kampüs Etkinlik Kayıt ve Katılım Sistemi	Kulüp etkinlikleri, öğrenci kayıtları, kontenjan ve katılım takibi.	GET filtreleme, POST kayıt, session, rol kontrolü, rapor, işlem günlüğü.
2. Randevu ve Sıra Yönetim Sistemi	Hizmet seçimi, uygun saat arama, randevu alma ve görevli onayı.	Saat çakışması kontrolü, görevli paneli, durum güncelleme, işlem günlüğü.
3. Mini E-Ticaret ve Sipariş Yönetim Sistemi	Ürün kataloğu, sepet, sipariş oluşturma ve yönetici ürün yönetimi.	Sepet mantığı, stok kontrolü, sipariş kalemleri, fiş dosyası.
4. Destek Talebi ve Bilgi Bankası Sistemi	Talep açma, mesajlaşma, durum takibi ve bilgi bankası araması.	Talep akışı, mesaj tablosu, GET filtreleri, rol kontrolü, işlem günlüğü.

1. Genel amaç ve öğrenme çıktıları

- HTML ile semantik sayfa yapısı, form alanları, label-input ilişkisi, tablo ve liste çıktıları oluşturmak.
- CSS ile sade, okunabilir ve mobilde bozulmayan bir arayüz geliştirmek.
- JavaScript ile form doğrulama, küçük hesaplamalar, filtreleme veya kullanıcıya anlık geri bildirim gibi istemci tarafı etkileşimler eklemek.
- PHP ile GET/POST verilerini almak, koşullar, döngüler, fonksiyonlar, include/require yapıları ve session kullanmak.
- PHP içinde değişkenleri ekrana yazdırırken echo kullanmak ve kullanıcıdan gelen veriyi güvenli şekilde göstermek.
- MySQL ile ilişkili tablolar tasarlamak; kayıt ekleme, listeleme, güncelleme ve silme işlemlerini PDO hazırlanmış sorgular ile yapmak.
- Dosyaya yazma işlemi kullanarak işlem günlüğü, basit rapor veya fiş dosyası üretmek.

2. Önerilen klasör yapısı

```
Örnek klasör yapısı
donem_projesi/
├── varliklar/
│   ├── css/style.css
│   └── js/app.js
├── ortak/
│   ├── db.php
│   ├── fonksiyonlar.php
│   ├── ust.php
│   └── alt.php
├── sayfalar/
│   ├── giris.php
│   ├── kayit.php
│   ├── panel.php
│   └── yonetim.php
├── veritabani/
│   ├── schema.sql
│   └── seed.sql
├── kayitlar/ veya fisler/
├── ekran_goruntuleri/
├── index.php
└── README.md
```

3. Genel değerlendirme rubriği

Kategori	Puan	Beklenen seviye
HTML ve CSS arayüzü	15	Semantik HTML, anlaşılır formlar, sade ve bozulmayan tasarım.
JavaScript kullanımı	10	Form doğrulama, ön izleme, uyarı veya filtreleme gibi anlamlı küçük etkileşimler.
PHP GET/POST ve sunucu tarafı doğrulama	20	GET filtreleri, POST işlemleri, hata mesajları, echo ile güvenli çıktı.
MySQL ve PDO kullanımı	20	İlişkili tablolar, hazırlanmış sorgular, doğru ekleme/listeleme/güncelleme/silme akışı.
Session, rol ve güvenlik kontrolleri	15	Giriş, yetki kontrolü, password_hash, yetkisiz erişimi engelleme.
Dosya işlemi ve proje organizasyonu	10	Log/fiş/rapor dosyası üretimi, düzenli klasör yapısı, açıklanabilir kod.
Demo ve açıklama	10	Çalışan demo, test kullanıcıları, veritabanı anlatımı, README ve ekran görüntüleri.

Proje Seçeneği 1 - Kampüs Etkinlik Kayıt ve Katılım Sistemi

Kulüp etkinlikleri için duyuru, kayıt, kontenjan ve katılım takibi uygulaması.

Düz hedef görünüm - Kampüs Etkinlik Kayıt ve Katılım Sistemi

Üst Menü	Etkinlikler	Giriş / Panel
Arama Alanı	Kategori seçimi + tarih + kelime arama	GET ile URL değiştir
Liste	Etkinlik adı, tarih, kontenjan, kayıt butonu	Detay bağlantısı
Panel	Öğrencinin kayıtları / yöneticinin etkinlik yönetimi	İşlem günlüğü

Beklenen ekran tasarımları

Not: Görseller birbir kopyalanacak tasarım değildir; sayfa akışını, veri alanlarını ve beklenen işlevleri gösteren örnek hedef ekranlardır.

Örnek ekran 1: Herkese açık etkinlik listesi, GET arama/filtre ve POST kayıt akışı.

The screenshot shows a web application interface for 'Kampüs Etkinlikleri'. The page has a blue header with the title and navigation links for 'Etkinlikler', 'Kulüpler', and 'Giriş'. Below the header, there is a section titled 'Yaklaşan Etkinlikler' with a subtitle 'GET filtreleme ile kategori, tarih ve kelime araması'. A search bar contains three input fields: 'Kategori' (Teknoloji), 'Tarih' (2026-05-22), and 'Anahtar kelime' (workshop). A blue 'Ara' button is next to the search bar. Below the search bar, there are three event cards. Each card has a title, date, category, and a 'Kontenjan' (quota) indicator. The first card is 'Web Workshop' (22 Mayıs • Teknoloji) with a quota of 26 / 40. The second is 'Kariyer Buluşması' (24 Mayıs • Kariyer) with a quota of 55 / 60. The third is 'Tasarım Semineri' (27 Mayıs • Sanat) with a quota of 18 / 30. Each card also has a 'Detay / Kayıt' button. At the bottom of the page, there are four links: 'HTML form', 'GET arama', 'POST kayıt', and 'MySQL liste'.

The screenshot shows a web application interface for event management. The top navigation bar includes 'Panel', 'Etkinlik Ekle', and 'Çıkış'. The main heading is 'Yönetici Kontrol Ekranı'. Below this, there are four summary cards: 'Toplam etkinlik 10', 'Bugünkü kayıt 37', 'Dolu etkinlik 2', and 'Log satır 124'. A table lists events with columns for 'Etkinlik', 'Tarih', 'Kapasite', 'Kayıt', and 'İşlem'. The 'İşlem' column contains links like 'Düzenle / Sil' and 'Dolu / Rapor'. A 'İşlem günlüğü' (Log) section shows a list of actions with timestamps. At the bottom, there are buttons for 'Yeni Etkinlik Ekle' and 'Rapor', and a footer with 'Session rol kontrolü', 'PDO CRUD', and 'file_put_contents log'.

Etkinlik	Tarih	Kapasite	Kayıt	İşlem
Web Workshop	22.05.2026	40	26	Düzenle / Sil
Kariyer Buluşması	24.05.2026	60	55	Düzenle / Sil
Tasarım Semineri	27.05.2026	30	18	Düzenle / Sil
PHP Mini Hackathon	29.05.2026	25	25	Dolu / Rapor

Hikaye / Senaryo

Üniversitedeki öğrenci kulüpleri etkinlik duyurularını tek bir sistemde paylaşmak istiyor. Öğrenciler etkinlikleri kategoriye, tarihe ve anahtar kelimeye göre arayacak; detay sayfasından kayıt olacak; kontenjan dolmuşsa sistem yeni kayıt almayacak. Kulüp yöneticisi etkinlik oluşturacak, kayıtları görecektir ve etkinlik günü katılım işaretleyecektir.

Öğrenme odağı

- Öğrenci ve yönetici rolleriyle oturum yönetimi kurmak.
- GET ile etkinlik arama ve filtreleme yapmak: kategori, tarih, anahtar kelime.
- POST ile etkinlik kaydı, iptal, geri bildirim ve yönetici ekleme/güncelleme işlemleri yapmak.
- Kontenjan kontrolü kurmak: kayıt sayısı kapasiteyi geçmemelidir.
- Etkinlik oluşturma, kayıt ve iptal işlemlerini bir metin dosyasına loglamak.

Beklenen sayfalar ve ekranlar

Ekran	Beklenen işlev
Ana sayfa	Yaklaşan etkinlikleri listeler ve arama/filtre formunu gösterir.
Etkinlik detay sayfası	Başlık, tarih, kategori, açıklama, kontenjan ve kayıt butonu içerir.
Kayıt / giriş / çıkış sayfaları	Kullanıcı hesabı oluşturur, giriş yapar ve session bilgisini yönetir.
Öğrenci paneli	Kayıt olunan etkinlikleri listeler; iptal ve geri bildirim formu içerir.
Yönetici paneli	Etkinlik ekleme, güncelleme, silme ve katılım işaretleme işlemlerini yapar.
Rapor sayfası	Etkinlik sayısı, toplam kayıt sayısı ve dolan etkinlikleri gösterir.

Minimum veri tabanı tasarımı

Tablo	Önerilen alanlar / ilişki
-------	---------------------------

Tablo	Önerilen alanlar / ilişki
kullanıcılar	id, ad, eposta, sifre_hash, rol, olusturma_tarihi
etkinlikler	id, baslik, kategori, tarih, aciklama, kapasite, yoneticisi_id
etkinlik_kayitlari	id, etkinlik_id, kullanıcı_id, kayıt_tarihi, durum
katilimler	id, etkinlik_id, kullanıcı_id, katildi_mi
geri_bildirimler	id, etkinlik_id, kullanıcı_id, puan, yorum

HTML, CSS, JavaScript, PHP ve MySQL kullanımı

Teknoloji	Beklenen kullanım
HTML	Formlar, etiketler, tablo/listeler ve etkinlik detay sayfa yapısı semantik olarak yazılmalıdır.
CSS	Basit, okunabilir, mobilde bozulmayan bir tasarım yapılmalıdır. Hazır tema yerine sade kendi düzeni tercih edilmelidir.
JavaScript	Form alanı boş kontrolü, tarih uyarısı ve kontenjan uyarısı gibi küçük etkileşimler eklenmelidir.
PHP	GET filtrelerini okumalı, POST kayıtlarını işlemeli, session kontrolü yapmalı ve echo ile dinamik çıktı üretmelidir.
MySQL	Etkinlik, kullanıcı ve kayıt tabloları ilişkili olmalı; PDO hazırlanmış sorgular kullanılmalıdır.
Dosya işlemi	etkinlik_log.txt içine etkinlik ekleme, kayıt ve iptal işlemleri yazılmalıdır.

Nasıl yapılacak? Önerilen geliştirme sırası

1. Veri tabanı tablolarını oluşturun ve farklı kategorilerde en az 10 örnek etkinlik ekleyin.
2. Kodlamaya başlamadan önce includes/db.php ve includes/fonksiyonlar.php dosyalarını hazırlayın.
3. Önce herkese açık etkinlik listesi ve etkinlik detay sayfalarını geliştirin.
4. Öğrenci kayıt/giriş/çıkış akışını session ile kurun.
5. Etkinliğe kayıt olma POST işlemini ve kontenjan kontrolünü yazın.
6. Yönetici etkinlik ekleme/güncelleme/silme sayfalarını hazırlayın.
7. Katılım işaretleme ve basit rapor sayfasını ekleyin.
8. Geçersiz veri, dolu etkinlik ve yetkisiz erişim durumlarını test edin.

Öğrenciden beklenen minimum teslim

- Öğrenci aynı etkinliğe iki kez kayıt olamaz.
- Kontenjanı dolan etkinlik anlaşılır bir uyarı gösterir ve yeni kayıt almaz.
- Yönetici sayfaları rol kontrolü olmadan açılmaz.
- Etkinlik listesi ve detayları MySQL verisinden dinamik üretilir.
- Demo sırasında en az bir işlem günlüğü satırı dosyaya yazılır.

Proje Seçeneği 2 - Randevu ve Sıra Yönetim Sistemi

Danışmanlık, klinik, kuaför, laboratuvar veya kampüs birimi için zaman aralığına dayalı rezervasyon uygulaması.

Düz hedef görünüm - Randevu ve Sıra Yönetim Sistemi

Üst Menü	Randevu Ara	Giriş / Görevli Paneli
Arama	Hizmet + görevli + tarih	GET ile sonuç listesi
Uygun Saatler	Saat, görevli, durum, randevu al butonu	POST formu
Görevli Paneli	Günlük sıra listesi, onay/red/tamamlandı	İşlem logu

Beklenen ekran tasarımları

Not: Görseller birebir kopyalanacak tasarım değildir; sayfa akışını, veri alanlarını ve beklenen işlevleri gösteren örnek hedef ekranlardır.

Örnek ekran 1: Uygun saat arama, randevu seçimi ve POST randevu formu.

Örnek ekran 2: Görevli paneli, günlük sıra listesi ve durum güncelleme ekranı.

Saat	Kişi	Hizmet	Durum	Görevli işlem	Not
09:00	Elif Demir	Danışmanlık	Onaylı	Tamamlandı	Geldi
10:00	Mert Kaya	Danışmanlık	Bekliyor	Onayla/Reddet	Gerekçe var
11:30	Ali Koç	Belge teslim	Bekliyor	Onayla/Reddet	Eksik belge
13:00	Zeynep Arı	Laboratuvar	İptal	Kapalı	Kullanıcı iptal

Hikaye / Senaryo

Bir kampüs birimi randevularını telefonla almak yerine web üzerinden yönetmek istiyor. Kullanıcı hizmet türünü, görevliyi ve tarihi seçerek uygun saatleri görür. Randevu talebi gönderir; görevli panelinde bu talebi onaylar, reddeder veya tamamlandı olarak işaretler. Aynı saat aralığına iki onaylı randevu verilemez.

Öğrenme odağı

- GET ile hizmet, görevli ve tarih araması yapmak.
- POST ile randevu oluşturma, iptal talebi ve durum güncelleme işlemleri yapmak.
- Saat çakışması önlemek ve uygunluk kontrolü kurmak.
- Görevli/yönetici rolüyle randevu onaylama veya reddetme işlemi yapmak.
- Günlük sıra listesi ve randevu durum paneli hazırlamak.

Beklenen sayfalar ve ekranlar

Ekran	Beklenen işlev
Ana arama sayfası	Hizmet, görevli ve tarih filtrelerini içerir.
Randevu alma sayfası	Seçilen saat, açıklama/gerekçe alanı ve gönder butonu içerir.
Kullanıcı paneli	Yaklaşan randevuları ve iptal talebi butonunu gösterir.
Görevli paneli	Günlük sıra, onay, red ve tamamlandı işlemlerini içerir.
Yönetici sayfası	Hizmet, görevli ve uygun saat yönetimi yapar.
İstatistik sayfası	Günlük randevu sayısı ve bekleyen talep sayısını gösterir.

Minimum veri tabanı tasarımı

Tablo	Önerilen alanlar / ilişki
kullanıcılar	id, ad, eposta, sifre_hash, rol
hizmetler	id, ad, açıklama, aktif_mi
gorevliler	id, kullanıcı_id, uzmanlık, aktif_mi
uygun_saatler	id, görevli_id, hizmet_id, tarih, baslangic_saati, bitis_saati
randevular	id, kullanıcı_id, uygun_saat_id, gerekce, durum, olusturma_tarihi

HTML, CSS, JavaScript, PHP ve MySQL kullanımı

Teknoloji	Beklenen kullanım
HTML	Arama formu, randevu formu, tablo halinde saat listesi ve panel ekranları hazırlanmalıdır.
CSS	Randevu listesi okunabilir olmalı; renk kullanımı az ve anlaşılır olmalıdır.
JavaScript	Boş alan kontrolü, tarih seçimi uyarısı ve seçilen saat vurgusu yapılmalıdır.
PHP	GET filtreleriyle saatleri listelemeli, POST ile randevu almalı ve durum güncellemelidir.
MySQL	Uygun saat ve randevu tabloları ilişkilendirilmeli; çakışma kontrolü sorgu ile yapılmalıdır.
Dosya işlemi	randevu_islemleri.txt dosyasına onay, red ve iptal işlemleri yazılmalıdır.

Nasıl yapılacak? Önerilen geliştirme sırası

1. Tabloları tasarlayın; hizmet, görevli ve uygun saat örnek verileri ekleyin.
2. GET arama sayfasını kurun ve uygun saatleri listeleyin.
3. Randevu alma formunu POST doğrulaması ve tekrar randevu kontrolüyle yazın.
4. Kullanıcı ve görevli giriş/session akışını oluşturun.
5. Görevli durum güncelleme işlemlerini geliştirin.
6. Bekleyen ve onaylanan randevular için panel sayaçları ekleyin.
7. JavaScript ile küçük doğrulama ve seçilen saat uyarısı ekleyin.
8. Dolu saat, eksik gerekçe ve yetkisiz sayfa erişimi durumlarını test edin.

Öğrenciden beklenen minimum teslim

- Onaylı bir randevunun bulunduğu saat tekrar alınamaz.
- Kullanıcı yalnızca kendi randevularını görebilir.
- Görevli durum güncellemesini yalnızca giriş yaptıktan sonra yapabilir.
- GET filtreleri URL içinde açıkça görünür.
- Demo sırasında işlem günlüğü dosyası değişir.

Proje Seçeneği 3 - Mini E-Ticaret ve Sipariş Yönetim Sistemi

Küçük okul mağazası, kitap/kırtasiye satışı veya kulüp ürünleri için katalog, sepet ve sipariş uygulaması.

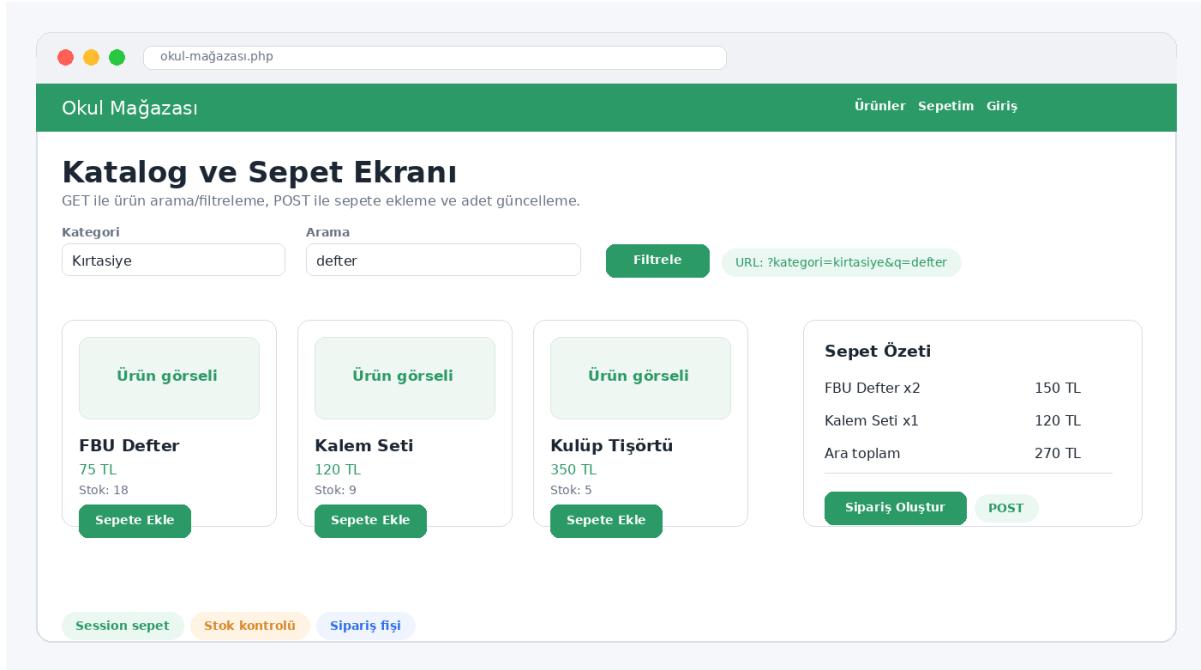
Düz hedef görünüm - Mini E-Ticaret ve Sipariş Yönetim Sistemi

Üst Menü	Katalog	Sepet / Giriş
Filtre	Kategori + kelime + sıralama	GET ile URL değişir
Ürün Listesi	Ürün adı, fiyat, stok, sepete ekle	POST ile sepet
Yönetici Paneli	Ürün ekle/güncelle, sipariş durumu	Fiş dosyası

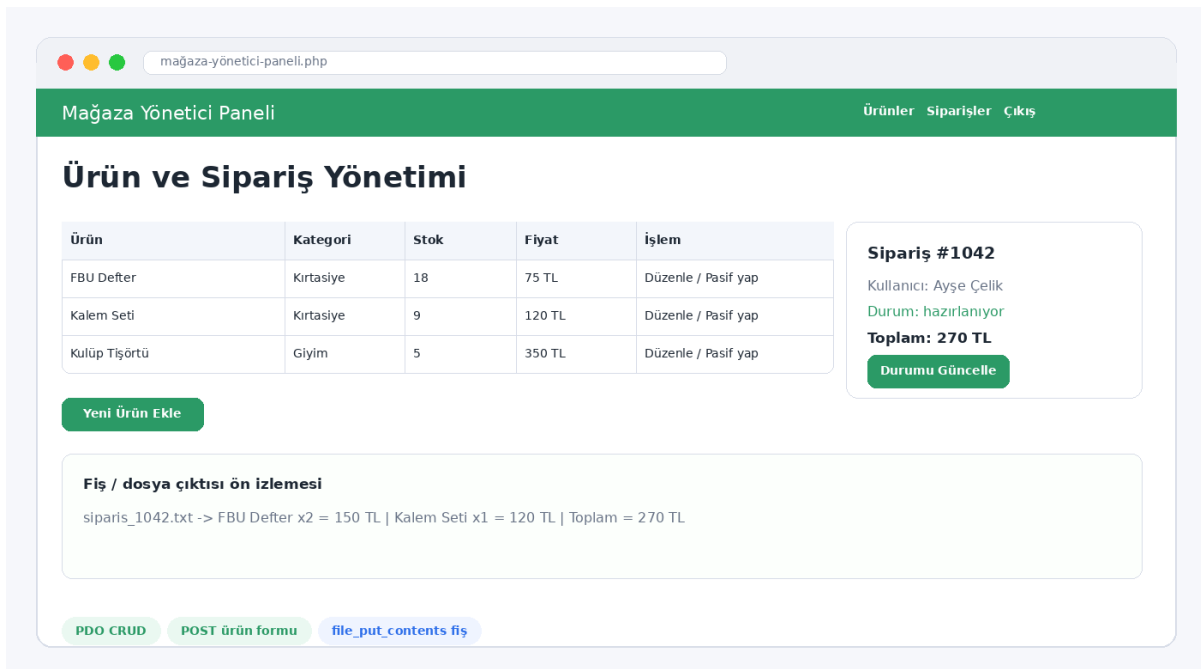
Beklenen ekran tasarımları

Not: Görseller birebir kopyalanacak tasarım değildir; sayfa akışını, veri alanlarını ve beklenen işlevleri gösteren örnek hedef ekranlardır.

Örnek ekran 1: Ürün kataloğu, GET filtreleme, sepet özeti ve sipariş butonu.



Örnek ekran 2: Ürün yönetimi, sipariş durumu ve dosyaya yazılan fiş ön izlemesi.



Hikaye / Senaryo

Kampüs mağazası ürünlerini çevrim içi listelemek ve öğrencilerin basit sipariş oluşturmasını sağlamak istiyor. Öğrenci ürünleri kategoriye göre gezer, arama yapar, sepete ekler ve gerçek ödeme entegrasyonu olmadan sipariş oluşturur. Yönetici ürün stoklarını ve sipariş durumlarını yönetir.

Öğrenme odağı

- GET ile kategori, arama ve sıralama kullanan ürün kataloğu hazırlamak.
- Session tabanlı veya veritabanı tabanlı sepet mantığı kurmak.
- POST ile sepete ekleme, adet güncelleme ve sipariş oluşturma işlemleri yapmak.
- Yönetici ürün ekleme/güncelleme/silme ve sipariş durumu yönetimi yapmak.
- Her sipariş için basit bir metin fişi üretmek.

Beklenen sayfalar ve ekranlar

Ekran	Beklenen işlev
Katalog sayfası	Ürün listesi, arama, kategori filtresi ve sıralama içerir.
Ürün detay sayfası	Açıklama, fiyat, stok ve sepete ekleme formu içerir.
Sepet sayfası	Ürünler, adet güncelleme, ara toplam ve sepet boşaltma işlemi içerir.
Sipariş tamamlama sayfası	Teslim bilgisi ve sipariş onayı formu içerir.
Kullanıcı sipariş geçmişi	Kullanıcının verdiği siparişleri ve durumlarını gösterir.
Yönetici sayfaları	Ürün, kategori, stok ve sipariş durumu yönetimi yapar.

Minimum veri tabanı tasarımı

Tablo	Önerilen alanlar / ilişki
kullanıcılar	id, ad, eposta, sifre_hash, rol
kategoriiler	id, ad, açıklama
urunler	id, kategori_id, ad, açıklama, fiyat, stok, aktif_mi
siparisler	id, kullanıcı_id, toplam_tutar, durum, teslim_adresi, olusturma_tarihi
siparis_ogeleri	id, siparis_id, urun_id, adet, birim_fiyat

HTML, CSS, JavaScript, PHP ve MySQL kullanımı

Teknoloji	Beklenen kullanım
HTML	Ürün kartları/listeleri, sepet tablosu, sipariş formu ve yönetici formları hazırlanmalıdır.
CSS	Katalog ve sepet ekranları sade ve okunabilir olmalıdır; aşırı karmaşık tasarım beklenmez.
JavaScript	Adet kontrolü, ara toplam ön izlemesi ve boş sepet uyarısı yapılmalıdır.
PHP	GET filtrelerini okumalı, POST sepet/sipariş işlemlerini yapmalı ve echo ile çıktı üretmelidir.
MySQL	Ürün, sipariş ve sipariş kalemleri ilişkili olmalı; stok kontrolü sunucu tarafında yapılmalıdır.
Dosya işlemi	Her sipariş için fisler/ klasöründe basit bir sipariş fişi oluşturulmalıdır.

Nasıl yapılacak? Önerilen geliştirme sırası

1. Ürün, kategori, sipariş ve sipariş kalemi tablolarını oluşturun; en az 12 ürün ekleyin.
2. GET arama/filtre ile katalog ve ürün detay sayfalarını geliştirin.
3. Sepet mantığını session veya sepet tablosu ile kurun.
4. Sipariş oluşturma POST akışını ve siparis_ogeleri kayıtlarını yazın.
5. Her sipariş için metin fişi üretin.
6. Yönetici ürün ekleme/güncelleme ve sipariş durumu sayfalarını hazırlayın.
7. JavaScript ile ara toplam ön izlemesi ve adet uyarısı ekleyin.
8. Stok yetersizliği, boş sepet ve geçersiz adet durumlarını test edin.

Öğrenciden beklenen minimum teslim

- Kullanıcı stoktan fazla ürün sipariş edemez.
- Siparisler ve siparis_ogeleri kayıtları tutarlı şekilde eklenir.
- Sepet ara toplamı ile sunucuda hesaplanan toplam uyumludur.
- Yönetici ürün ekleyebilir, düzenleyebilir ve pasif hale getirebilir.
- Fiş dosyası oluşur ve doğru toplam tutarı içerir.

Proje Seçeneği 4 - Destek Talebi ve Bilgi Bankası Sistemi

Kampüs IT veya teknik destek birimi için talep açma, yanıtlaşma, durum takibi ve bilgi bankası uygulaması.

Düz hedef görünüm - Destek Talebi ve Bilgi Bankası Sistemi

Üst Menü	Yardım / Bilgi Bankası	Talep Aç / Giriş
Arama	Kategori + durum + öncelik	GET ile filtre
Talep Detayı	Mesaj geçmişi + yanıt formu	POST ile yanıt
Destek Paneli	Tüm talepler, atama, durum güncelleme	İşlem logu

Beklenen ekran tasarımları

Not: Görseller birebir kopyalanacak tasarım değildir; sayfa akışını, veri alanlarını ve beklenen işlevleri gösteren örnek hedef ekranlardır.

Örnek ekran 1: Bilgi bankası araması ve yeni destek talebi formu.

Örnek ekran 2: Destek ekibi paneli, talep listesi, konuşma geçmişi ve durum işlemleri.

No	Başlık	Durum	Öncelik	Atanan
#352	Wi-Fi kopuyor	Açık	Yüksek	Mehmet
#353	E-posta sorunu	Bekliyor	Normal	Ayşe
#354	Yazıcı hatası	Kapalı	Düşük	Selin

Hikaye / Senaryo

Öğrenciler ve personel bilgisayar, hesap, internet veya laboratuvar sorunlarını e-posta yerine tek bir destek sisteminden iletmek istiyor. Kullanıcı destek talebi açar, durumunu takip eder ve destek ekibi yanıt yazdığında görüşmeyi aynı sayfada görür. Sık sorulan konular bilgi bankasında aranabilir.

Öğrenme odağı

- POST formlarıyla destek talebi oluşturmak ve mesaj yanıtı eklemek.
- GET ile durum, kategori ve öncelik filtreleri kurmak.
- talep_mesajlari tablosu ile konuşma geçmişi göstermek.
- Destek ekibi/yönetici rolüyle yanıt, atama ve durum güncelleme işlemleri yapmak.
- Bilgi bankası araması ve makale ekleme/güncelleme işlemleri hazırlamak.

Beklenen sayfalar ve ekranlar

Ekran	Beklenen işlev
Yardım ana sayfası	Bilgi bankası araması ve talep açma bağlantısı içerir.
Talep oluşturma sayfası	Kategori, öncelik, başlık ve açıklama alanları içerir.
Kullanıcı paneli	Kullanıcının taleplerini, durumlarını ve son yanıtlarını gösterir.
Talep detay sayfası	Tüm konuşma geçmişi ve yanıt formu içerir.
Destek paneli	Tüm talepler, filtreler, atama ve durum güncelleme işlemleri içerir.
Yönetici sayfası	Kategoriler ve bilgi bankası makalelerini yönetir.

Minimum veri tabanı tasarımı

Tablo	Önerilen alanlar / ilişki
kullanıcılar	id, ad, eposta, sifre_hash, rol
kategoriler	id, ad, açıklama
talepler	id, kullanıcı_id, kategori_id, baslik, oncelik, durum, atanan_id, olusturma_tarihi
talep_mesajlari	id, talep_id, kullanıcı_id, mesaj, olusturma_tarihi
bilgi_makaleleri	id, kategori_id, baslik, icerik, yayinlandi_mi

HTML, CSS, JavaScript, PHP ve MySQL kullanımı

Teknoloji	Beklenen kullanım
HTML	Talep formu, filtre formu, mesaj listesi ve bilgi bankası sayfaları semantik hazırlanmalıdır.
CSS	Destek listesi ve konuşma alanı okunabilir olmalıdır; karmaşık tasarım beklenmez.
JavaScript	Mesaj karakter sayacı, silme/onay uyarısı ve boş alan kontrolü eklenmelidir.
PHP	GET filtreleri, POST talep/yanıt formları, session rol kontrolü ve echo ile çıktı üretimi yapılmalıdır.
MySQL	Talep ve mesaj tabloları ilişkili olmalı; kullanıcı yalnızca kendi taleplerini görebilmelidir.
Dosya işlemi	talep_islemleri.txt dosyasına talep açma, yanıt ve durum değişiklikleri yazılmalıdır.

Nasıl yapılacak? Önerilen geliştirme sırası

1. Tabloları oluşturun; kategori, kullanıcı ve bilgi makalesi örnek verileri ekleyin.
2. Bilgi bankası listeleme/arama sayfasını GET ile geliştirin.
3. Talep oluşturma formunu sunucu tarafı doğrulama ile yazın.
4. Giriş/session akışını ve kullanıcı panelini kurun.
5. Talep detayında konuşma geçmişi ve yanıt POST işlemini geliştirin.
6. Destek panelinde durum güncelleme ve atama işlemlerini oluşturun.
7. JavaScript ile karakter sayacı ve onay mesajları ekleyin.
8. Yetkisiz erişim, boş açıklama ve kapalı talebe yanıt durumlarını test edin.

Öğrenciden beklenen minimum teslim

- Normal kullanıcı yalnızca kendi taleplerini görür; destek/yönetici tüm talepleri görebilir.
- Her yeni talep için ilk açıklama talep_mesajlari tablosuna da kaydedilir.
- Destek ekibi açık talepleri filtreleyebilir ve durum güncelleyebilir.
- Bilgi bankası makaleleri GET aramasıyla bulunabilir.
- Demo sırasında talep işlem günlüğü dosyası güncellenir.