

WEB PROGRAMLAMA I ■ HAFTA 14

PHP ile Sunucu Taraflına Giriş

Neden sunucu gerekli? ■ CGI'dan PHP'ye ■ PHP temelleri ■ Form işleme ■ Sunucu taraflı doğrulama ■ Web-2'ye köprü

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

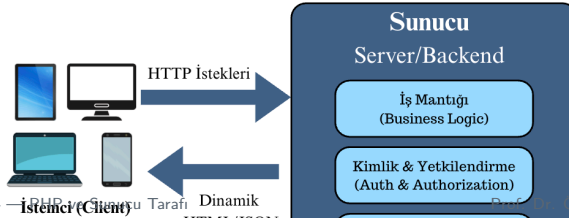
İstemci Tarafının Sınırına Geldik

Geçen hafta `data/kitaplar.json` dosyasından veri çıktık. Peki bu dosyayı **kim günceller**?

⚠ Dikkat

Cevap: Bir geliştirici, **elle**. Yeni bir kitap eklemek için dosyayı açıp düzenlemesi ve sunucuya yüklemesi gerekiyor. **İşte bu noktada istemci tarafı JavaScript'in sınırlarına ulaştık.**

Tarayıcı, **güvenilmeyen ve geçici** bir ortamdır. Gerçek bir uygulama için **merkezi, güvenilir ve kalıcı bir beyne** — yani bir **sunucu uygulamasına** ihtiyaç duyarız.



- ▶ Sunucu tarafı programlamanın neden zorunlu olduğunu dört başlıkta açıklayabilecek,
- ▶ CGI'dan PHP'ye giden tarihsel gelişimi özetleyebilecek,
- ▶ Yerel bir PHP ortamı kurup `.php` dosyası çalıştırabilecek,
- ▶ PHP'nin temel sözdizimini (değişken, koşul, döngü, dizi) kullanabilecek,
- ▶ `$_POST` ve `$_GET` ile form verisi işleyebilecek,
- ▶ Sunucu tarafı doğrulamanın neden vazgeçilmez olduğunu gösterebilecek,
- ▶ PHP'den JSON üretip `fetch` ile tüketebilecek,
- ▶ Web-2 dönemine hangi konularla devam edeceğini bilebileceksiniz.

- 1 Neden Sunucu?
- 2 Sunucu Tarafının Tarihi
- 3 PHP Temelleri
- 4 Form İşleme
- 5 Yapay Zeka ve Sunucu Güvenliği
- 6 Dönem Değerlendirmesi



Neden Sunucu?

BÖLÜM 1

Dört Temel Neden

1. Kalıcı Veri Saklama

Binlerce kitabı, yüzlerce kullanıcıyı tek bir `.json` dosyasında saklamak imkânsızdır. Bu veriler **veritabanında** tutulur. İstemci JavaScript, **güvenlik nedeniyle bir veritabanına asla doğrudan bağlanamaz.**

2. Kullanıcı Yönetimi ve Güvenlik

Kimlik doğrulama ve yetkilendirme **asla istemcide olamaz** — tarayıcıdaki JS kodu görülebilir ve değiştirilebilir. Biri `if (kullanici == 'admin')` satırını **kolayca atlayabilirdi.**

3. Merkezi İş Mantığı

Sepet toplamı, indirim kodları, ödeme işlemi, öneri algoritması... Bu işlemler **tüm kullanıcılar için tutarlı** olmalı ve **güvenli bir ortamda** çalışmalıdır.

4. Servis Entegrasyonu

Ödeme geçitleri, sosyal medya API'leri, kargo servisleri... Bunlar **API anahtarı** gerektirir. Gizli anahtarlar **asla istemci JavaScript'ine konulmamalıdır.**

Kısaca: Sunucu tarafı; verinin **kalbi**, iş mantığının **beyni** ve uygulamanın **güvenlik kalkanıdır.**

Somut Örnek: Neden İstemcide Olmaz?

✗ İstemcide (güvensiz)

```
// js/main.js - HERKES GOREBİLİR
const API_ANAHTARI = "sk_live_51H8x...";

function girisYap(kullanici, sifre) {
  if (kullanici === "admin" && sifre === "1234") {
    yoneticisiPaneliniAc();
  }
}

function indirimUygula(kod) {
  if (kod === "BOOKVERSE50") {
    return fiyat * 0.5;
  }
}
```

Kullanıcı ne yapabilir?

- ▶ API anahtarını çalar
- ▶ Şifreyi kaynak koddan okur
- ▶ Konsolda `yoneticisiPaneliniAc()` çağırır
- ▶ Tüm indirim kodlarını görür

Altın kural: Tarayıcıya gönderilen her şey **kullanıcının kontrolündedir**. Güvenlik kararları **sadece sunucuda** verilir.

✓ Sunucuda (güvenli)

```
<?php
// giris.php - KULLANICI GOREMEZ
$apiAnahtari = getenv("STRIPE_KEY");

$kullanici = $_POST["kullanici"];
$sifre      = $_POST["sifre"];

$kayit = veritabanindanGetir($kullanici);

if (password_verify($sifre, $kayit["sifre_hash"])) {
  $_SESSION["yetki"] = $kayit["rol"];
  echo json_encode(["durum" => "basarili"]);
} else {
  http_response_code(401);
  echo json_encode(["hata" => "Hatali giris"]);
}
```

Sunucu Tarafının Tarihi

BÖLÜM 2

CGI: Sunucuda Kod Çalıştırmanın İlk Yolu

Web'in ilk yıllarında sunucular sadece **statik HTML dosyalarını** sunabiliyordu. Geliştiriciler sordu: *"Bu dosyayı göndermeden önce sunucuda bir program çalıştırıp, onun ürettiği çıktıyı göndersek olmaz mı?"*

Tanım

CGI (Common Gateway Interface) bir teknoloji veya dil değil, bir **protokoldür**. Web sunucusunun gelen isteği harici bir programa nasıl ileteceğini ve o programdan gelen çıktıyı nasıl geri göndereceğini tanımlar.

Perl Dönemi

1990'ların ortasında web'deki dinamik içeriğin büyük kısmı — ziyaretçi sayacıları, misafir defterleri, form işleyiciler — **Perl ile yazılmış CGI script'leriydi**.

CGI'in Sorunu

Gelen **her istek için yeni bir işlem (process)** başlatıyordu. Yüksek trafikte sunucu kaynakları (bellek, CPU) **hızla tükeniyordu**.

PHP ve ASP: İki Devrimci Fikir

CGI'in performans sorununu çözmek için 1990'ların ortasında **PHP** ve **ASP** ortaya çıktı.

⚡ 1. Devrim: Modül Olarak Çalışma

Kod, **web sunucusunun kendi içinde** bir modül olarak çalışıyordu. Her istek için **yeni işlem başlatma maliyeti ortadan kalktı** — performans **dramatik** biçimde arttı.

📄 2. Devrim: HTML İçine Gömme

Programlama kodunu **doğrudan HTML'in içine** yazmak. Statik bir sayfanın **sadece belirli kısımlarını** dinamik hale getirmek **inanılmaz kolaylaştı**.

```
1 <!DOCTYPE html>
2 <html>
3 <head><title>PHP Örneği</title></head>
4 <body>
5   <h1>BookVerse</h1>
6   <p>Bugünün tarihi: <?php echo date("d.m.Y"); ?></p>
7   <p>Sunucu: <?php echo $_SERVER["SERVER_SOFTWARE"]; ?></p>
8 </body>
9 </html>
```

Bugün Sunucu Tarafı Seçenekleri

Teknoloji	Dil	Not
PHP	PHP	Web'in %75'inden fazlası hâlâ PHP kullanıyor (WordPress)
Node.js	JavaScript	Aynı dili hem istemcide hem sunucuda kullanma
ASP.NET Core	C#	Kurumsal projeler, Microsoft ekosistemi
Django / Flask	Python	Veri bilimi ve yapay zeka entegrasyonu kolay
Spring Boot	Java	Büyük ölçekli kurumsal sistemler
Ruby on Rails	Ruby	Hızlı prototipleme

İpucu

Neden bu derste PHP?

- ▶ Kurulumu **en kolay** olan (XAMPP ile 5 dakika)
- ▶ Kodu **HTML içine gömülebildiği** için geçiş yumuşak
- ▶ Türkiye'de **yaygın** — staj ve iş ilanlarında karşınıza çıkar



PHP Temelleri

BÖLÜM 3

Ortam Kurulumu

PHP dosyaları **bir sunucu tarafından işlenmelidir** — Live Server yetmez.

Seçenek 1: XAMPP

apachefriends.org adresinden indirin.
Apache + PHP + MySQL bir arada gelir.
Dosyalarınızı **htdocs/** klasörüne koyun,
http://localhost/bookverse adresinden açın.

>_ Seçenek 2: Yerleşik Sunucu

```
# PHP kuruluysa, proje klasöründe:  
php -S localhost:8000  
  
# Tarayıcıda:  
# http://localhost:8000
```

Tek komut, ek kurulum yok. **Geliştirme için idealdir.**

```
1 <?php  
2 // test.php - kurulum kontrolu  
3 echo "PHP çalışıyor! Surum: " . phpversion();  
4 ?>
```

⚠ Dikkat

Dosyayı çift tıklarsanız tarayıcı PHP kodunu **düz metin olarak** gösterir — veya hiç göstermez. Mutlaka **http://localhost/...** adresi üzerinden açın.

PHP Sözdizimi — JavaScript ile Karşılaştırma

```
1 <?php
2 // Degiskenler - $ ile baslar, tip belirtilmez
3 $siteAdi = "BookVerse";
4 $yil      = 2026;
5 $aktif    = true;
6 $fiyat    = 89.90;
7
8 // Metin birlestirme - NOKTA ile (JS'te + idi)
9 $mesaj = $siteAdi . " projesine hos geldiniz!";
10
11 // Cift tirnak icinde degisken CALISIR
12 echo "Site: $siteAdi, Yil: $yil";
13 echo 'Site: $siteAdi';           // tek tirnakta CALISMAZ
14
15 // Ciktilama
16 echo "Merhaba";                 // yazdirir
17 print_r($dizi);                 // dizi/nesne yapisini gosterir
18 var_dump($degisken);            // tip + deger gosterir (hata ayiklama)
19 ?>
```

Koşullar, Döngüler ve Diziler

```
1 <?php
2 // Kosullar - JavaScript ile neredeyse AYNI
3 if ($puan >= 9)      { echo "Mukemmel"; }
4 elseif ($puan >= 7) { echo "Iyi"; }
5 else                { echo "Ortalama"; }
6
7 // Donguler
8 for ($i = 1; $i <= 5; $i++) { echo "Sayfa $i"; }
9 while ($sayac < 3)         { $sayac++; }
10
11 // Indeksli dizi
12 $kitaplar = ["Simyaci", "Kürk Mantolu", "Tutunamayanlar"];
13 echo $kitaplar[0];
14 echo count($kitaplar);
15
16 // Iliskisel dizi (JS'teki nesnenin karsiligi)
17 $kitap = [
18     "ad"      => "Simyaci",
19     "yazar"   => "Paulo Coelho",
20     "fiyat"   => 89.90
21 ];
22 echo $kitap["ad"];
```

HTML İçine Gömülü PHP

```
1 <?php
2 $kitaplar = [
3     ["ad" => "Simyaci",          "yazar" => "Coelho", "fiyat" => 89.90],
4     ["ad" => "Kürk Mantolu",     "yazar" => "Ali",   "fiyat" => 45.00],
5     ["ad" => "Tutunamayanlar",   "yazar" => "Atay",   "fiyat" => 120.00]
6 ];
7 ?>
8 <!DOCTYPE html>
9 <html lang="tr">
10 <head><meta charset="UTF-8"><title>BookVerse</title></head>
11 <body>
12     <h1>Kitaplarımız (<? = count($kitaplar) ?> adet)</h1>
13
14     <div class="kitap-listesi">
15         <?php foreach ($kitaplar as $k): ?>
16             <article class="kitap-kart">
17                 <h3><? = htmlspecialchars($k["ad"]) ?></h3>
18                 <p><? = htmlspecialchars($k["yazar"]) ?></p>
19                 <p><? = number_format($k["fiyat"], 2) ?> TL</p>
20             </article>
21         <?php endforeach; ?>
22     </div>
23 </body>
```



Form İşleme

BÖLÜM 4

\$_POST ve \$_GET

HTML form

```
<form action="kayit.php" method="POST">
  <label for="ad">Adiniz:</label>
  <input type="text" id="ad" name="ad" required>

  <label for="mail">E-posta:</label>
  <input type="email" id="mail" name="mail" required>

  <button type="submit">Kayıt Ol</button>
</form>
```

kayit.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $ad = trim($_POST["ad"] ?? "");
    $mail = trim($_POST["mail"] ?? "");

    echo "Merhaba " . htmlspecialchars($ad);
}
?>
```

Özellik

\$_GET

\$_POST

Veri nerede?

URL'de görünür

İstek gövdesinde

Yer imine eklenebilir mi?

Evet

Hayır

Boyut sınırı

Var (~2000 karakter)

Pratikte yok

Kullanım

Arama, filtreleme, sayfalama

Kayıt, giriş, şifre

Hafta 3'e dönüş: Formun `method` özneliği burada **hangi PHP değişkenine** düşeceğini belirler.

Sunucu Tarafı Doğrulama — Vazgeçilmez

```
1 <?php
2 $hatalar = [];
3
4 if ($_SERVER["REQUEST_METHOD"] === "POST") {
5     $ad = trim($_POST["ad"] ?? "");
6     $mail = trim($_POST["mail"] ?? "");
7
8     // 1) Bos kontrolu
9     if ($ad === "") { $hatalar[] = "Ad alani zorunludur."; }
10    if (mb_strlen($ad) < 3) { $hatalar[] = "Ad en az 3 karakter olmalidir."; }
11
12    // 2) Format kontrolu
13    if (!filter_var($mail, FILTER_VALIDATE_EMAIL)) {
14        $hatalar[] = "Gecerli bir e-posta adresi giriniz.";
15    }
16
17    // 3) Sonuc
18    if (empty($hatalar)) {
19        // veritabanina kaydet, e-posta gonder, vb.
20        echo "Kaydiniz alindi!";
21    }
22 }
```

PHP'den JSON Üretmek: API'nin Doğuşu

```
1 <?php
2 // api/kitaplar.php
3 header("Content-Type: application/json; charset=utf-8");
4
5 $kitaplar = [
6     ["id" => 1, "ad" => "Simyaci",          "yazar" => "Coelho", "fiyat" => 89.90],
7     ["id" => 2, "ad" => "Kurk Mantolu",     "yazar" => "Ali",    "fiyat" => 45.00],
8     ["id" => 3, "ad" => "Tutunamayanlar",   "yazar" => "Atay",   "fiyat" => 120.00]
9 ];
10
11 // Arama parametresi varsa filtrele
12 $sterim = $_GET["q"] ?? "";
13 if ($sterim !== "") {
14     $kitaplar = array_values(array_filter($kitaplar, function ($k) use ($sterim) {
15         return mb_stripos($k["ad"], $sterim) !== false;
16     }));
17 }
18
19 echo json_encode($kitaplar, JSON_UNESCAPED_UNICODE);
20 ?>
```

Uygulama: BookVerse Bülten Formu

```

1 <?php
2 // bulten.php
3 $mesaj = "";
4 $tur = "";
5
6 if ($_SERVER["REQUEST_METHOD"] === "POST") {
7     $mail = trim($_POST["mail"] ?? "");
8
9     if (!filter_var($mail, FILTER_VALIDATE_EMAIL)) {
10         $mesaj = "Gecerli bir e-posta adresi giriniz.";
11         $tur = "hata";
12     } else {
13         // Basit kalicilik: dosyaya ekle (gercekte veritabani kullanilir)
14         file_put_contents("data/aboneler.txt", $mail . PHP_EOL, FILE_APPEND);
15         $mesaj = "Kaydiniz alindi. Tesekkurler!";
16         $tur = "basari";
17     }
18 }
19 ?>
20 <form action="bulten.php" method="POST">
21     <label for="mail">E-posta Adresiniz:</label>
22     <input type="email" id="mail" name="mail" required>
23     <button type="submit">Kayit Ol</button>
24
25     <?php if ($mesaj !== ""): ?>
26         <p class="mesaj" <? = $tur ?>><? = htmlspecialchars($mesaj) ?></p>
27     <?php endif; ?>
28 </form>

```

PHP Fonksiyonları ve Dosya Dahil Etme

```
1 <?php
2 // Fonksiyon tanimi - varsayılan parametre destekli
3 function kdvliFiyat(float $fiyat, float $oran = 0.20): float {
4     return $fiyat * (1 + $oran);
5 }
6
7 echo kdvliFiyat(89.90);           // 107.88
8 echo kdvliFiyat(89.90, 0.10);    // 98.89
9
10 // Dizi fonksiyonlari - JavaScript'e çok benzer
11 $fiyatlar = [89.90, 45.00, 120.00];
12 echo array_sum($fiyatlar);        // toplam
13 echo count($fiyatlar);            // adet
14 $zamli = array_map(fn($f) => $f * 1.2, $fiyatlar); // map
15 $ucuz = array_filter($fiyatlar, fn($f) => $f < 100); // filter
16
17 // Dosya dahil etme - kod tekrarını önler
18 require "parcalar/ustbilgi.php"; // yoksa PROGRAMI DURDURUR
19 include "parcalar/yanpanel.php"; // yoksa sadece uyarır
20 require_once "config.php";       // sadece bir kez dahil et
21 ?>
```

Şablon Parçalarıyla Kod Tekrarını Önlemek

Her sayfada aynı <header> ve <footer> kodunu tekrar yazmak yerine **parçalara bölün**.

```
bookverse/  
|-- index.php  
|-- kitaplar.php  
|-- iletisim.php  
|-- parcalar/  
|   |-- ustbilgi.php  
|   |-- menu.php  
|   |-- altbilgi.php  
|-- api/  
|   |-- kitaplar.php  
|-- data/  
    |-- kitaplar.json
```

```
<?php  
// index.php  
$sayfaBasligi = "Ana Sayfa";  
require "parcalar/ustbilgi.php";  
?>  
  
<main>  
  <h1>BookVerse'e Hos Geldiniz</h1>  
  <!-- sayfaya OZEL icerik -->  
</main>  
  
<?php require "parcalar/altbilgi.php"; ?>
```

İpucu

Kazanım: Menüye yeni bir link eklemek için artık **tek bir dosyayı** düzenliyorsunuz — 5 HTML dosyasını değil. Bu, Hafta 5'te CSS için öğrendiğimiz **"tek noktadan değişiklik"** ilkesinin HTML karşılığıdır.

Sunucu Tarafında Güvenlik: Üç Altın Kural

Kural	Uygulama
1. Girdiyi asla güvenme	Gelen her <code>\$_POST</code> / <code>\$_GET</code> değerini doğrula ve temizle .
2. Çıktıyı kaçışla	Ekrana basarken <code>htmlspecialchars(\$deger)</code> kullan — XSS önlenir.
3. Sorguyu hazırla	Veritabanı sorgusuna değişken yapıştırma ; <i>prepared statement</i> kullan — SQL Injection önlenir.

✗ Açık

```
$ad = $_POST["ad"];  
echo "Merhaba " . $ad;  
  
$sorgu = "SELECT * FROM kitaplar  
        WHERE ad = '$ad'";
```

Saldırgan ad alanına

' OR '1' ='1 yazarsa **tüm tabloyu** dökabilir.

✓ Güvenli

```
$ad = trim($_POST["ad"] ?? "");  
echo "Merhaba " . htmlspecialchars($ad);  
  
$stmt = $pdo->prepare(  
    "SELECT * FROM kitaplar WHERE ad = ?"  
);  
$stmt->execute([$ad]);
```

Şifre saklama: Asla düz metin veya `md5` kullanmayın. `password_hash()` ve `password_verify()`.

PHP Hata Ayıklama

```
1 <?php
2 // GELISTIRME ortamında hataları göster
3 ini_set("display_errors", 1);
4 error_reporting(E_ALL);
5
6 // Degisken icerigini incele
7 var_dump($_POST);           // tip + deger
8 print_r($kitaplar);         // okunakli dizi ciktisi
9
10 echo "<pre>";                // <pre> icinde daha okunakli
11 print_r($kitaplar);
12 echo "</pre>";
13 ?>
```

Hata

Sebep

Sayfa **tamamen boş**

İsözdizimi hatası. `display_errors` açık mı?

Undefined array key "ad"

Form alanında `name` eksik veya yanlış

Headers already sent

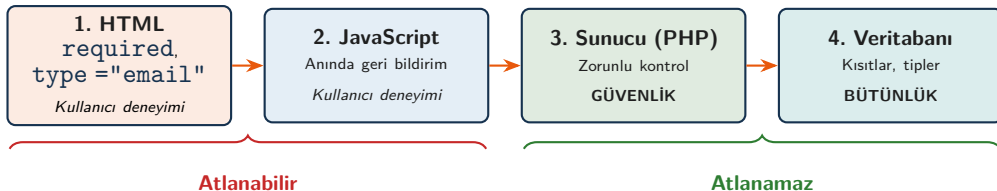
`header()` öncesinde `echo` yapılmış

Oturum Yönetimi (session)

HTTP **durumsuzdur** (stateless) — her istek birbirinden bağımsızdır. Peki kullanıcının “giriş yapmış” olduğunu nasıl hatırlarız?

```
1 <?php
2 session_start();           // HER sayfanin EN BASINDA
3
4 // Giris basarili olunca
5 $_SESSION["kullanici_id"] = 42;
6 $_SESSION["ad"]           = "Ayse";
7 $_SESSION["rol"]          = "uye";
8
9 // Baska bir sayfada kontrol
10 session_start();
11 if (!isset($_SESSION["kullanici_id"])) {
12     header("Location: giris.php");
13     exit;
14 }
15 echo "Hos geldiniz, " . htmlspecialchars($_SESSION["ad"]);
16
17 // Cikis
18 session_destroy();
19 ?>
```

İstemci ve Sunucu Doğrulaması: Katmanlı Savunma



İpucu

1 ve 2 numaralı katmanlar kullanıcıya yardım eder — formu doldururken anında geri bildirim verir, sunucuya gereksiz istek gitmesini önler. **Ama 3 ve 4 numaralı katmanlar olmadan sisteminiz güvensizdir.** Saldırgan tarayıcıyı hiç kullanmadan, doğrudan `curl` ile sunucunuza istek gönderebilir.

Doğrulama Atlatma: Kendiniz Deneyin

Terminalden, tarayıcıyı **tamamen atlayarak** sunucunuza istek gönderin:

```
# Formu doldurmadan, dogrudan sunucuya POST istegi
curl -X POST http://localhost:8000/bulten.php \
  -d "mail=gecersizmail"

# JavaScript dogrulamasi CALISMADI - sunucu ne yapiyor?
```

Sunucu Doğrulaması Yoksa

Geçersiz e-posta **veritabanına kaydedilir**. Saldırgan bir döngüyle **binlerce sahte kayıt** oluşturabilir.

Sunucu Doğrulaması Varsa

```
filter_var($mail,
FILTER_VALIDATE_EMAIL) kontrolü isteği
reddeder — tarayıcı kullanılmış mı kullanılmamış
mı fark etmez.
```

Bu deney dönemin en önemli dersidir: İstemci tarafı doğrulama bir **nezakettir**; sunucu tarafı doğrulama bir **zorunluluktur**.

Yapay Zeka ve Sunucu Güvenliđi

BÖLÜM 5

Sunucu Kodunda Yapay Zeka: En Yüksek Risk

İstemci tarafında bir hata **kötü bir deneyim** yaratır. Sunucu tarafında bir hata **veri sızıntısı** yaratır.

🦴 Yapay Zekanın Ürettiği Klasik Açıklar

- ▶ **SQL Injection** — kullanıcı verisini sorguya doğrudan ekleme
- ▶ **XSS** — `htmlspecialchars()` kullanmama
- ▶ Şifreyi **düz metin** saklama (`md5` önerme)
- ▶ Dosya yükleme kontrolü yapmama
- ▶ Hata mesajında **veritabanı bilgisi** sızdırma
- ▶ API anahtarını **kodun içine** yazma

🛡 Prompt Kısıtlarınız

"Veritabanı sorgularında **prepared statement** (hazırlanmış ifade) kullan." "Tüm çıktıları **htmlspecialchars** ile kaçışla." "Şifre için **password_hash** kullan, `md5/sha1` kullanma." "Gizli anahtarları **ortam değişkeninden** oku."

Bu dönem öğrendiğiniz en önemli şey: Yapay zeka **çalışan** kod üretir — **güvenli** kod ürettiğini siz doğrularsınız.

Yapay Zeka İstemi (Prompt) — PHP Güvenlik İncelemesi

Asagidaki PHP kodunu GUVENLIK acisinden denetle.

[KODUNUZU YAPISTIRIN]

Su acik turlerini kontrol et. Her biri icin [GUVENLI] / [RISKLI] / [KRITIK] etiketi ver:
1) XSS - kullanıcı verisi htmlspecialchars ile kacsiliyor mu? 2) SQL Injection - sorgular prepared statement mi? 3) Girdi dogrulama - `_POST/_GET` degerleri kontrol ediliyor mu? 4) Hata mesajlari - teknik detay sizdiriyor mu? 5) Gizli bilgiler - API anahtari/sifre kodun icinde mi? 6) Dosya islemleri - yol (path) manipulasyonuna acik mi?

Her [RISKLI] ve [KRITIK] icin: - Saldirgan ne yapabilir? (somut ornek) - Duzeltme icin HANGI fonksiyon kullanilmali?

KURAL: Duzeltilmis tam kodu VERME. Sadece degistirilecek satiri "eski -> yeni" olarak goster.

Yapay Zeka İstemi (Prompt) — Öğrenme Planı Oluşturma

Bir donem boyunca su konulari ogrendim:

HTML5 (semantik etiketler, formlar, erisilebilirlik), CSS (secici/cascade, box model, Flexbox, Grid, duyarli tasarim), JavaScript (degiskenler, fonksiyonlar, dizi metotlari, DOM, olaylar, async/await, fetch, JSON), PHP giris (form isleme, sunucu tarafli dogrulama)

Gelecek donem "Web Programlama II" dersinde sunlari gorecegim: Modern JavaScript (ES6+), React, paket yoneticileri, REST API, Node.js/Express, veritabanlari, Next.js, Docker, AWS, CI/CD

GOREV: 1) Bu iki donem arasindaki TATIL doneminde ne calismaliyim? 4 haftalik bir plan yap (haftada 10 saat). 2) Her hafta icin: konu, kaynak onerisi, kucuk bir proje fikri 3) Ogrendiklerimden HANGI konuyu pekistirmeden gecmemeliyim? Nedenini acikla. 4) Bir "kontrol listesi" ver: Web-2'ye baslamadan once rahatca yapabilmem gereken 10 sey. Turkce yaz. Gercekci ol - asiri iddiali plan yapma.

PHP'de Veritabanına Kısa Bir Bakış

Web-2'de detaylı göreceğiz — şimdilik **nasıl görüldüğünü** bilelim.

```
1 <?php
2 // Baglanti (PDO - modern ve guvenli yontem)
3 $pdo = new PDO("mysql:host=localhost;dbname=bookverse;charset=utf8mb4",
4               "kullanici", "sifre",
5               [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
6
7 // OKUMA - hazirlanmis ifade (prepared statement)
8 $stmt = $pdo->prepare("SELECT * FROM kitaplar WHERE tur = ? ORDER BY puan DESC");
9 $stmt->execute(["roman"]);
10 $kitaplar = $stmt->fetchAll(PDO::FETCH_ASSOC);
11
12 // YAZMA
13 $ekle = $pdo->prepare("INSERT INTO kitaplar (ad, yazar, fiyat) VALUES (?, ?, ?)");
14 $ekle->execute([$ad, $yazar, $fiyat]);
15
16 // JSON olarak dondur - Hafta 13'teki fetch bunu tuketir
17 header("Content-Type: application/json; charset=utf-8");
18 echo json_encode($kitaplar, JSON_UNESCAPED_UNICODE);
19 ?>
```

Terim	Anlam	Terim	Anlam
Client-side	İstemci (tarayıcı) tarafı	\$_POST	Gövdeyle gelen form verisi
Server-side	Sunucu tarafı	\$_GET	URL'den gelen veri
CGI	Sunucuda program çalıştırma protokolü	echo	Çıktı verme
PHP	Hypertext Preprocessor	htmlspecialchars	XSS kaçıışı
Authentication	Kimlik doğrulama	filter_var	Girdi doğrulama
Authorization	Yetkilendirme	password_hash	Güvenli şifre saklama
Session	Sunucudaki oturum verisi	Prepared statement	SQL Injection koruması
Business logic	İş mantığı	PDO	PHP veritabanı katmanı



Dönem Değerlendirmesi

BÖLÜM 6

14 Haftada Neler Yaptık?

⚡ Hafta 1–4: Yapı

Web'in temelleri, HTTP, DNS ▪ HTML belge iskeleti ▪ Tablolar ve formlar ▪ Semantik yapı, erişilebilirlik, SEO

🖌 Hafta 5–9: Görünüm

CSS seçiciler ve cascade ▪ Box model ▪ Tipografi ve renk ▪ Flexbox ▪ Grid ve duyarlı tasarım

⚡ Hafta 10–13: Davranış

JavaScript temelleri ▪ Fonksiyonlar ve dizi metotları ▪ DOM ve olaylar ▪ Asenkron JS, fetch, JSON

☰ Hafta 14: Sunucu

Neden sunucu gerekli ▪ PHP temelleri ▪ Form işleme ▪ Sunucu tarafı doğrulama ▪ PHP'den JSON API

Sonuç: Boş bir klasörden başlayıp; semantik, erişilebilir, duyarlı, interaktif ve **veriyle beslenen** bir web uygulaması olan **BookVerse**'ü inşa ettiniz.

Web-2'de Sizi Neler Bekliyor?

Hafta	Konu	Neyi çözer?
1-2	Modern JavaScript (ES6+), modüller	Büyük projeleri düzenleme
3-4	React ve bileşen mimarisi	Arayüzü yeniden kullanılabilir parçalara bölme
5-6	Paket yöneticileri, build araçları	Bağımlılık yönetimi, optimizasyon
7-8	REST API tasarımı ve tüketimi	İstemci-sunucu sözleşmesi
9-10	Node.js, Express, veritabanı	Gerçek bir backend yazma
11-12	Next.js ve bütünleşik mimariler	SSR, performans, SEO
13-14	Docker, AWS, CI/CD	Uygulamayı dünyaya açma

İpucu

En önemli müjde: Web-2'de öğreneceğiniz her şey, bu dönem öğrendikleriniz **üzerine** inşa edilir. React'in **JSX**'i **HTML**'dir; bileşen stilleri **CSS**'tir; mantık **JavaScript**'tir. *Temeliniz sağlamsa, Web-2 çok daha kolay geçecek.*

Erişilebilirlik

Her hafta bir erişilebilirlik konusu işledik. **Bu bir “ekstra özellik” değil**, mesleğin etik ve yasal bir gereğidir.

Güvenlik

XSS deneyini kendi gözünüzle gördünüz. **“Çalışıyor” ile “güvenli” aynı şey değildir.**

Yapay Zeka Okuryazarlığı

14 hafta boyunca yapay zekayı **yasaklamadık — doğrulamayı** öğrendik. Prompt yazmayı, kısıt koymayı, çıktıyı sınamayı ve **halüsinasyonu tanımayı** öğrendiniz.

Teşhis Alışkanlığı

DevTools'u her hafta kullandık. **Tahmin etmeyin — ölçün.**

*“Bu kitap sadece bir kod rehberi değil, aynı zamanda bir mühendisin **problem çözme yaklaşımını** kazanmanız için bir haritadır.”*

- 1 İstemci tarafı JavaScript'in **asla yapamayacağı** dört işi yazınız.
- 2 CGI'nin en büyük performans sorunu neydi? PHP bunu nasıl çözdü?
- 3 PHP'de " ve ' tırnakları arasındaki fark nedir?
- 4 Aşağıdaki kodda **iki güvenlik açığı** vardır. Bulunuz:

```
$ad = $_POST["ad"];  
echo "Merhaba " . $ad;  
$sorgu = "SELECT * FROM kitaplar WHERE ad = '$ad'";
```

- 5 HTML'deki `required` önitelik ve JavaScript doğrulaması varken sunucu tarafı doğrulama neden hâlâ zorunludur?
- 6 `$_GET` ile `$_POST` arasındaki farkı ve hangisini ne zaman kullanacağınızı açıklayınız.
- 7 `htmlspecialchars()` fonksiyonu, Hafta 12'de öğrendiğimiz hangi JavaScript özelliğine karşılık gelir?

Laboratuvar Föyü 14

- ▶ PHP ortam kurulumu
- ▶ Temel sözdizimi alıştırmaları
- ▶ HTML içine gömülü PHP ile kitap listesi
- ▶ Bülten formunun sunucuda işlenmesi
- ▶ **Doğrulama atlatma deneyi**
- ▶ PHP'den JSON API üretme
- ▶ Yapay zeka ile güvenlik denetimi

Dönem Projesi

BookVerse projenizin son hali:

- ▶ Semantik ve erişilebilir HTML
- ▶ Duyarlı CSS (Flexbox + Grid)
- ▶ İnteraktif JavaScript (arama, filtre, sepet)
- ▶ `fetch` ile veri yükleme
- ▶ PHP ile form işleme

Teslim: Final sınavı haftası

Tatil önerisi: Kendi ilgi alanınızda **küçük bir proje** yapın — film listesi, tarif defteri, ders programı. BookVerse'te öğrendiklerinizi **sıfırdan** uygulamak, en hızlı pekiştirme yöntemidir.

Dönem Tamamlandı. Teşekkürler!

Web Programlama II'de görüşmek üzere

Modern JavaScript ▪ React ▪ Node.js ▪ REST API ▪ Next.js ▪ Docker ▪ AWS

Prof. Dr. Özal YILDIRIM ▪ Fırat Üniversitesi, Yazılım Mühendisliği Bölümü