

Site Yönetim Otomasyonu

Gülnehal Erushlu, Fatma Nilay Süzer, Cüneyt Şendur

Bilişim Sistemleri Mühendisliği, Teknoloji Fakültesi, Kocaeli Üniversitesi

gulni.erushlu@gmail.com, fatmanilaysuzer@gmail.com, cuneytsendur4@gmail.com

231307064

231307069

221307002

Özet

Bu proje, site yöneticileri ile sakinlerin iletişim ve yönetim ihtiyaçlarını karşılamak amacıyla geliştirilen, PHP programlama dili ve MySQL veritabanı kullanılarak XAMPP geliştirme ortamında yapılandırılmış web tabanlı bir site yönetim otomasyon sistemini sunmaktadır. İlgili verilerin saklanması için ilişkisel bir veritabanı oluşturulmuş ve kullanıcı etkileşimini sağlayan dinamik bir web arayüzü tasarlanmıştır.

Sistem üzerinden yöneticiler duyuru ekleyebilmekte ve sakinlerle ilgili işlemleri gerçekleştirebilmektedir. Sakinler ise eklenen duyuruları görüntüleyebilmekte ve çeşitli başvuru işlemlerini sistem aracılığıyla yapabilmektedir. Bu proje, apartman ve site gibi toplu yaşam alanlarında iletişimi kolaylaştırmayı ve yönetsel süreçleri dijital ortamda daha verimli hale getirmeyi amaçlamaktadır.

Abstract

This project presents the design and implementation of a web-based site management automation system developed using the PHP programming language and MySQL database within the XAMPP development environment. The system is designed to meet the communication and administrative needs of both site managers and residents. A relational database was created to store relevant data, and a dynamic web interface was developed to enable user interaction. The platform allows administrators to post announcements and manage resident-related operations, while residents can view announcements and submit various requests through the system. The project aims to facilitate efficient communication and streamline administrative workflows within residential communities.

1. Giriş

Site yönetiminde yaşanan aidat ödemelerindeki aksaklıklar, sakin değişikliği ve takibinin düzenli yapılamaması, site sakinlerinin öneri, talep ve şikayetlerinin hızlı bir şekilde iletilmemesi üzerine bu problemlerin daha hızlı çözülmesi için bu projeye ihtiyaç duyulmuştur.

Bu proje, site yöneticileri ile site sakinlerinin arasındaki iletişimi sağlama amaçlı yapılmış bir web sitesidir.

2. Web Sitesinin Oluşturulması

Web sitesi, XAMPP⁵ kullanılarak geliştirilmiştir; bu sayede PHP dosyaları ile MySQL veritabanı birlikte çalışabilecek şekilde entegre edilmiştir.

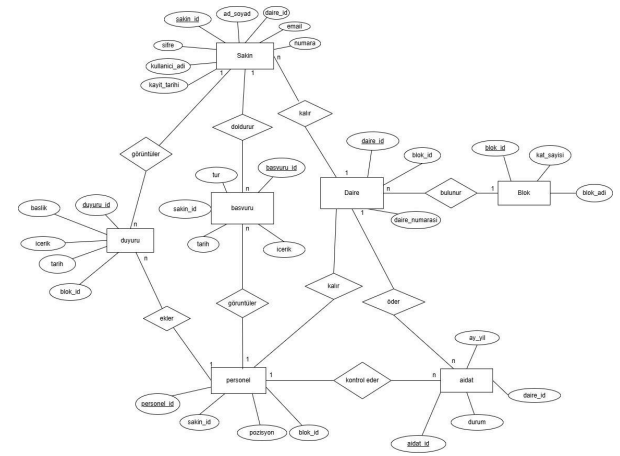
2.1. Veri Tabanının Oluşturulması

Veritabanı ve tabloların oluşturulmasında MySQL kullanılmış, bu işlemler phpMyAdmin arayüzü üzerinden görsel olarak gerçekleştirilmiştir. Veri tabanı üzerinde⁴

- sakinler
- bloklar
- daireler
- personeller
- duyurular
- başvurular
- aidatlar

tabloları oluşturulmuştur. Bunlar arasındaki ilişkiler ER diyagramında gösterilmiştir.

2.1.1. ER Diyagramı⁸



2.2. Uygulama Kodlarının Yazılması

Uygulama kodları, PHP ile sunucu tarafı işlemlerini gerçekleştirirken; HTML ve CSS kullanılarak kullanıcı dostu bir arayüz tasarlandı. Bu yapı sayesinde hem işlevsel hem de görsel olarak bütüncül bir web sitesi ortaya konmuştur

2.2.1. PHP

Projede sunucu tarafı tüm işlemler PHP programlama dili kullanılarak gerçekleştirilmiştir. PHP sayesinde kullanıcı girişleri, veri işleme, oturum yönetimi, veritabanı etkileşimleri ve dinamik içerik sunumu gibi işlevler başarıyla sağlanmıştır. Uygulamanın arka planındaki tüm işlem mantığı PHP dili ile kodlanmış ve sistemin dinamik şekilde çalışması sağlanmıştır³.

2.2.1.1 MySQLi

Veritabanı işlemleri sırasında güvenlik sağlamak için MySQLi kütüphanesi ile prepare(), bind_param() ve execute() gibi fonksiyonların kullanıldığı prepared statement yapısıyla SQL sorguları gerçekleştirilmiştir. Bu yapı sayesinde websitesinde SQL enjeksiyon saldırılarına karşı bir koruma sağlanmıştır.

```
$sql = "SELECT p.personel_id, s.kullanici_adi
FROM personel p
JOIN sakinler s ON p.sakin_id = s.sakin_id
WHERE s.kullanici_adi = ? AND s.sifre = ? AND p.pozisyon = 'Yonetici'";

$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $kullanici_adi, $sifre);
$stmt->execute();
$result = $stmt->get_result();
```

Şekil 1: yonetici_giris.php

```
if ($hedef_blok_id === null) {
    $sql = "INSERT INTO duyurular (baslik, icerik, tarih, hedef_blok_id) VALUES (?, ?, ?, NULL)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("sss", $baslik, $icerik, $tarih);
} else {
    $sql = "INSERT INTO duyurular (baslik, icerik, tarih, hedef_blok_id) VALUES (?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("sssi", $baslik, $icerik, $tarih, $hedef_blok_id);
}
```

Şekil 2: duyuru_ekle.php

2.2.1.2 Oturum (Session) Yönetimi

Kullanıcıların güvenli ve sürekli bir şekilde sisteme giriş yapabilmesi ve kullanıcıya özel işlemler gerçekleştirilebilmesi amacıyla oturum (session) yönetimi uygulanmıştır.

Kullanıcı doğrulama ve giriş/çıkış işlemleri PHP \$_SESSION yapısı ile, kullanıcı oturumu başlatma işlemi ise session_start() komutu ile yapılmıştır. Ayrıca session_start() komutu ile daha sonra kullanıcı oturumu kontrol edilerek yetkisiz erişim engellenmiştir. Bu yöntemle sakin girişi ve yönetici girişi kontrolü yapılmaktadır.

Güvenli giriş ve işlem takibi için ise PHP'nin \$_SESSION mekanizması kullanılarak kullanıcıya özel içerik gösterimi ve oturum yönetimi gerçekleştirilmiştir.

```
session_start();
if (!isset($_SESSION["yonetici"])) {
    header("Location: yonetici_giris.php");
    exit();
}
?>
```

Şekil 3: yonetici_panel.php

```
session_start();
include("baglanti.php");

if (!isset($_SESSION["sakin"])) {
    header("Location: sakin_giris.php");
    exit();
}

$mesaj = "";
$sakin_kadi = $_SESSION["sakin"];
```

Şekil 4: sakin_basvuru.php

2.2.1.3 Fonksiyonel panel yapısı

Yönetici yetkisine sahip kullanıcılar için PHP ile modüler olarak tasarlanmış fonksiyonel kontrol paneli üzerinden duyuru ekleme, sakin düzenleme, aidat takibi ve başvuruların görüntülenmesi işlemleriyle site yönetimi sağlanmaktadır.

- Duyuru Ekleme

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $baslik = $_POST["baslik"];
    $icerik = $_POST["icerik"];
    $tarih = date("Y-m-d H:i:s");

    $hedef_blok_id_raw = $_POST["hedef_blok_id"];
    $hedef_blok_id = ($hedef_blok_id_raw == "NULL") ? null : (int)$hedef_blok_id_raw;

    if ($hedef_blok_id == null) {
        $sql = "INSERT INTO duyurular (baslik, icerik, tarih, hedef_blok_id) VALUES (?, ?, ?, NULL)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("sss", $baslik, $icerik, $tarih);
    } else {
        $sql = "INSERT INTO duyurular (baslik, icerik, tarih, hedef_blok_id) VALUES (?, ?, ?, ?)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("sssi", $baslik, $icerik, $tarih, $hedef_blok_id);
    }

    if ($stmt->execute()) {
        $duyuru_basari = true;
    } else {
        $hata = "Hata: " . $stmt->error;
    }
}
```

Şekil 5: *duyuru_ekle.php*

- Sakin Düzenleme

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $ad_soyad = $_POST["ad_soyad"];
    $email = $_POST["email"];
    $numara = $_POST["numara"];
    $kullanici_adi = $_POST["kullanici_adi"];
    $daire_id = $_POST["daire_id"];

    $stmt = $conn->prepare("UPDATE sakinler SET ad_soyad = ?, email = ?, numara = ?, kullanici_adi = ?, daire_id = ? WHERE sakin_id = ?");
    $stmt->bind_param("sssiid", $ad_soyad, $email, $numara, $kullanici_adi, $daire_id, $sakin_id_duzenlenecek);

    if ($stmt->execute()) {
        echo "<script>alert('Sakin bilgileri @callendi. '); window.location.href='sakinleri_listele.php';</script>";
        exit();
    } else {
        echo "Hata: " . $stmt->error;
    }

    $stmt->close();
}
?>
```

Şekil 6: *sakin_duzenle.php*

- Aidat Takibi

```
if ($SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['daire_id'], $_POST['ay_yil'], $_POST['durum'])) {
    $daire_id = $_POST['daire_id'];
    $ay_yil = $_POST['ay_yil'];
    $durum = $_POST['durum'];

    // Kayıt varsa güncelle, yoksa ekle
    $check_sql = "SELECT * FROM aidatlar WHERE daire_id=? AND ay_yil=?";
    $stmt = $conn->prepare($check_sql);
    $stmt->bind_param("is", $daire_id, $ay_yil);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // Kayıt varsa güncelle
        $update_sql = "UPDATE aidatlar SET durum=? WHERE daire_id=? AND ay_yil=?";
        $update = $conn->prepare($update_sql);
        $update->bind_param("sis", $durum, $daire_id, $ay_yil);
        $update->execute();
    } else {
        // Kayıt yoksa ekle
        $insert_sql = "INSERT INTO aidatlar (daire_id, ay_yil, durum) VALUES (?, ?, ?)";
        $insert = $conn->prepare($insert_sql);
        $insert->bind_param("iss", $daire_id, $ay_yil, $durum);
        $insert->execute();
    }
}
```

Şekil 7: *aidat_takibi.php*

2.2.2 HTML ve CSS

Bu çalışmada web sayfasının hem içeriğini hem de görünümünü oluşturmak için HTML ve CSS birlikte kullanılmıştır.

Proje, kullanıcı dostu ve modern bir web arayüzü sunmak üzere HTML5 ile yapılandırılmıştır. Form elemanları (input, textarea, select vb.) dahil olmak üzere tüm kullanıcı giriş bileşenleri HTML5 etiketleri kullanılarak oluşturulmuştur.¹

CSS3 ile ise arayüz bileşenlerinin stil ve düzeni tanımlanmıştır. Stil teknikleri uygulanmış; mavi-yeşil renk paleti üzerinden tutarlı bir görsel tema oluşturulmuştur. Modern CSS teknikleri kullanılarak responsive ve tutarlı bir kullanıcı deneyimi sağlanmıştır. Böylece arayüz, hem işlevsel hem de performans açısından optimize edilmiştir.²

```

body {
  background-color: #defafa;
  font-family: 'Segoe UI', sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  flex-direction: column;
}

.container {
  text-align: center;
}

.menu {
  display: flex;
  flex-direction: column;
  gap: 20px;
}

.menu-item {
  text-decoration: none;
  padding: 15px 25px;
  border: 1px solid #003d3d;
  border-radius: 15px;
  background-color: #d6f0f0;
  color: #003d3d;
  font-size: 20px;
  transition: 0.3s ease;
  display: flex;
  align-items: center;
  justify-content: space-between;
  min-width: 300px;
}

.login-box {
  background-color: #defafa;
  border: 6px solid #003f3f;
  border-radius: 40px;
  padding: 30px;
  width: 300px;
  box-sizing: border-box;
  margin: 0 auto;
}

```

Şekil 8: style.css

```

<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <title>Duyurular</title>
  <style>
    body {
      background-color: #e3f6f6;
      font-family: 'Segoe UI', sans-serif;
      padding-top: 50px;
    }
    .duyuru {
      background-color: white;
      border: 2px solid #033c3c;
      border-radius: 15px;
      margin: 20px auto;
      padding: 20px;
      width: 70%;
    }
    .duyuru h3 {
      color: #033c3c;
    }
    .duyuru small {
      color: #555;
    }
    .button {
      margin-top: 20px;
      display: inline-block;
      padding: 10px 30px;
      border: 1px solid #033c3c;
      border-radius: 30px;
      background-color: white;
      font-size: 16px;
      color: #033c3c;
      text-decoration: none;
    }
    .button:hover {
      background-color: #d3f1f1;
    }
    .top-right {
      position: fixed;
      right: 30px;
      top: 20px;
      z-index: 100;
    }
  </style>
</head>
<body>

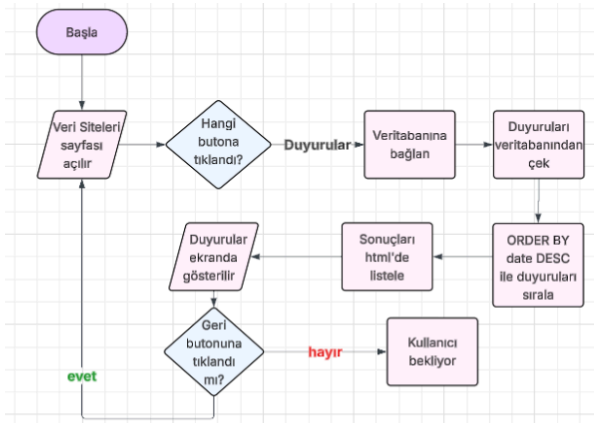
```

Şekil 9: duyurular.php

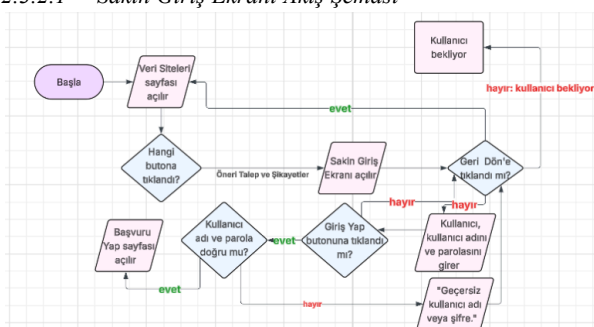
2.3. Akış Şemaları

Ana sayfada ‘duyurular’, ‘öneri talep ve şikayetler’, ‘yönetici girişi’ butonları bulunmaktadır. Alt başlık olarak verilen akış şemaları bu üç butonla açılan sayfalar ve butonları doğrultusunda şekillenmiştir.⁷

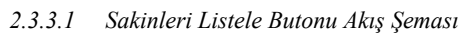
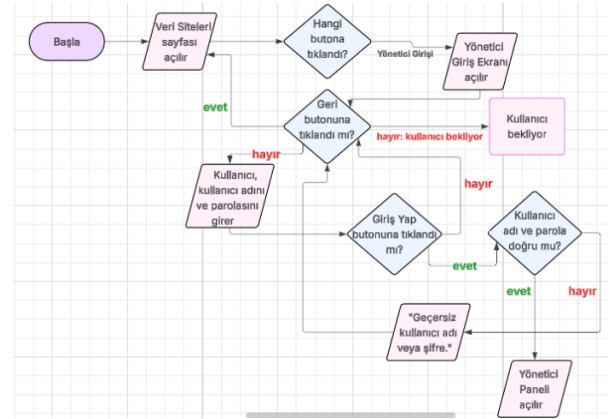
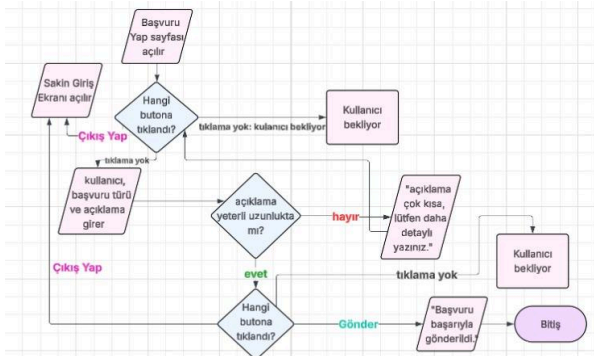
2.3.1. Duyurular butonu akış şeması

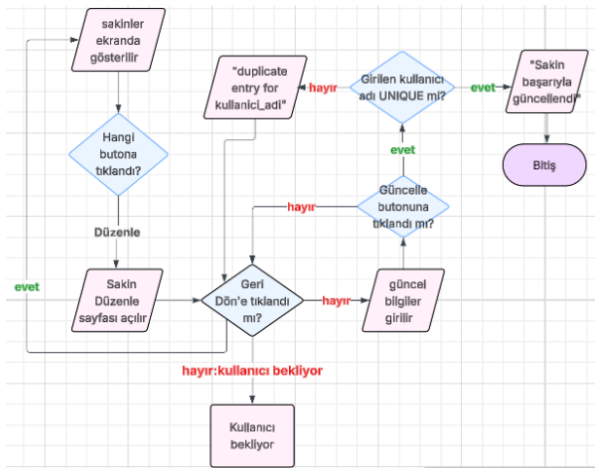


2.3.2. Öneri Talep ve Şikayetler butonu akış şeması

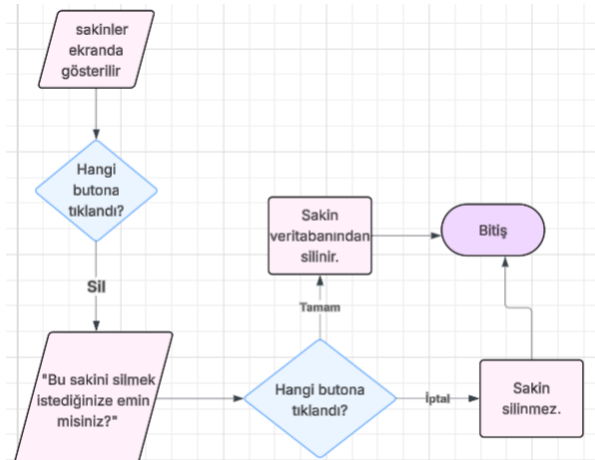


2.3.2.2 Başvuru Yap Ekranı Akış Şeması

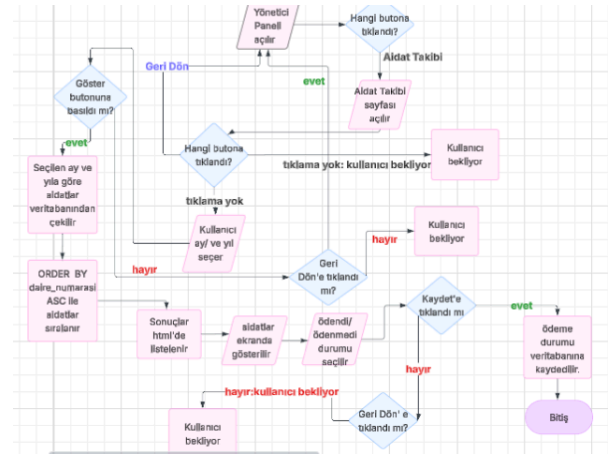




2.3.3.1.3 Sil Butonu Akış Şeması



2.3.3.1.4 Geri ve Çıkış Yap Butonu Akış Şeması



2.3.3.3 Duyuru Ekle Butonu Akış Şeması

[illegible]

```

graph TD
    A[Yönetici Paneli açılır] --> B{Hangi butona tıklandı?}
    B -- Çıkış Yap --> C[Yönetici Giriş Ekranı açılır]
    C --> D([Bitiş])
    B --> E{Geri butonuna tıklandı mı?}
    E -- evet --> A
    E -- hayır --> F[Kullanıcı bekliyor]
    F --> G[Öneri Talep ve Şikayet Görüntüleme]
    G --> H[Veritabanına bağlan]
    H --> I[Öneri, talep ve şikayetleri veritabanından çek]
    I --> J[ORDER BY date DESC ile öneri talep ve şikayetleri sırala]
    J --> K[Sonuçları html'de listele]
    K --> L[Öneri, talep ve şikayetler ekranda gösterilir]
  
```

arařtırmalar yaparak ve deneme-yanılma yöntemiyle bu sorunları çözmeyi başardık.

3.1. Normalizasyon Sorunu

Projenin ilk tasarım aşamasında, daireler tablosunda blok_adi alanı doğrudan ENUM olarak tanımlanmıştı. Bu yüzden blok bilgisiyle ilişkili verilerin ayrı bir tabloda tutulmaması nedeniyle veri tekrarına ve tutarsızlığa açık hale geliyordu.

Bu sorunu çözmek için veritabanı 5N normalizasyon kurallarına uygun olarak, bloklar adında ayrı bir tablo tanımlandı. Daireler tablosundaki blok_adi kaldırılarak yerine blok_id yabancı anahtar eklendi. Böylece her daire bir blokla ilişkisel olarak eşleştirilmiş oldu. Aynı yöntemle sakinler, aidatlar, duyurular ve personel tabloları da ilgili daire_id veya blok_id ile ilişkilendirildi.

PHP kodlarındaki SQL sorguları daha sade hale getirildi; örneğin bir yöneticinin sadece kendi bloğundaki sakinleri, aidatları veya başvuruları görmesi mümkün hale geldi.⁶

3.2. Veritabanı Bağlantı Sorunu

İlk başta, sistemin veritabanına bağlanmasını sağlayan kodları yazarken bağlantı kurulamıyordu. Sonradan fark edildi ki, veritabanı kullanıcı adı veya şifresi yanlış yazılmış. Doğru bilgiler girildikten sonra bağlantı başarılı şekilde sağlandı.

3.3. Giriş Yapınca Sayfaların Açılmaması

Kullanıcı giriş yaptıktan sonra bazı sayfalarda hata alınıyordu. Bunun nedeni, sayfanın başında oturum başlatılmasının (session) unutulmasıydı. Sayfanın başına "session_start();" eklenerek bu sorun çözüldü.

3.4. Kodlar Doğruyken Verilerin Gelmemesi

Duyuru formundan bilgi gönderildiğinde, veriler PHP tarafında gelmiyordu. Sorunun kaynağı, formun içinde name değerlerinin yazılmamasından kaynaklıydı. Form elemanlarına name="icerik" gibi tanımlar ekledikten sonra veri düzgün şekilde gelmeye başladı.

3.5. Yetkisiz Kişilerin Yönetici Paneline Girebilmesi

Başta, herkes yönetici sayfalarına girebiliyordu. Bu durumun önüne geçmek için, oturumu kontrol eden session_start() komutu yazıldı. Artık sadece oturumu açık olan ve yönetici yetkisi olan kullanıcılar bu sayfalara erişebiliyor duruma geldi.

3.6. Dosya Dahil Etme Sorunu

Bazı dosyaları projeye eklerken yanlış klasör yolu yazıldığı için hata alınıyordu. Doğru dosya yolu yazılınca sistem sorunsuz çalıştı.

4. Kaynaklar

- [1] <https://www.w3schools.com/html/default.asp>
- [2] <https://www.w3schools.com/css/default.asp>
- [3] <https://www.w3schools.com/php/>
- [4] <https://www.w3schools.com/mysql/>
- [5] <https://www.geeksforgeeks.org/how-to-install-xampp-on-windows/>
- [6] <https://elifsirin.medium.com/normalizasyon-nedir-%C3%B6rnekli-anlat%C4%B1m-d252a07622>
- [7] <https://www.lucidchart.com>
- [8] <https://draw.io/>

YouTube. (2024). *PHP MySQL CRUD Tutorial*.

GitHub. (2024). Açık kaynak PHP projeleri ve örnek sistemler. <https://github.com>