

**T.C.**

**BARTIN ÜNİVERSİTESİ**

**MÜHENDİSLİK, MİMARLIK VE TASARIM FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

|  |  |
| --- | --- |
| Ev Otomasyon Sistemi  **HAZIRLAYAN** | |
| Seyfullah Kurt  Beytullah Aziz Yapar | 20010310026  20010310028 |
|  | |
|  | |
| **DANIŞMAN**  **DR. ÖĞR. ÜYESİ Hakan Öcal** | |
|  | |

**BSM302 – YAZILIM MÜHENDİSLİĞİ**

**2023 – 2024 BAHAR DÖNEMİ**

## Giriş

Günümüzde ev işlerinde zamandan tasarruf kazanmak veya güvenliği artırmak amacı ile birçok akıllı ev aletlerinin ve İOT cihazların gelişmesiyle birlikte akıllı otomasyon sistemleri büyük önem kazanmıştır. Bunun için geliştirilen çözümler o kadar verimlidir ki birçok evin, iş yerinin ve ortak yaşam alanlarının vazgeçilmezi haline gelmiştir. Bu durum önemli bir güvenlik açığını beraberinde getirmektedir çünkü otomasyon sistemlerini oluşturan firmalar genellikle oluşturulan otomasyon sistemiyle ilgili tüm verileri kendi data warehouse’larında tutmaktadırlar. Bunun sebebi sisteme internet üzerinden erişimi kolaylaştırmakta, şirketin devamlı servis maliyetlerini düşürmekte ve kullanıcılara abonelik sistemini teşvik emektedir fakat bu data warehouse’ların büyüklüğü ve içindeki verilerin önemi siber saldırılara karşı adeta cazibe merkezi haline gelmekte ve saldırı yüzey alanını genişletmektedir. Bu da kişisel veri veya otomasyon sistemi üzerindeki hakimiyet için oldukça büyük bir güvenlik açığı oluşturmaktadır. Buna karşın bazı büyük çaplı işletmeler kendi altyapılarını oluşturma gibi alternatif çözümler üretmekte veya bu işletmeler için çözümler üreten girişimler bulunmaktadır fakat bireysel kullanıcılar bu alternatifleri oluşturmakta oldukça zorlanmaktadır. Bu durum yeterli bilgi birikimine, uğraşacak zamana sahip olmama veya alternatif çözüm üreten ürünlerin yaygın olmaması gibi sebeplerden kaynaklanmaktadır. İşte bu yüzden Ev Otomasyon Sistemi projesi ile otomasyon sisteminin data base’ini yerel sistem içerisinde tutarak bu büyük güvenlik açığından kurtulmak hedeflenmektedir.

Bu projede Python ve PHP-HTML dilleri kullanılarak, rasberry pi cihazında linux tabanlı bir işletim sistemi üzerinde çalışacak olan dinamik bir otomasyon kontrol sistemi tasarlanılacak ve MySQL ile oluşturulacak sistemin veritabanı hazırlanacaktır. Python dili ile sistemin linuxta çalışacak frameworku hazırlanılacak. PHP-HTML ile kullanıcının sisteme tarayıcı ile port üzerinden ve ip üzerinden bağlanabileceği arayüz tasarlanacaktır. Sistem uzaktan erişim için statik ip adresine sahip olmalıdır.

## Gerekliliklerin Analizi

1. **Donanım Gereksinimleri:**

* Raspberry Pi: Otomasyon sisteminin çalışacağı platform.
* SD Kart: Raspberry Pi işletim sistemi ve projeye ait dosyaların depolanması için minimum 16GB.
* Sensörler ve Aktüatörler: Çeşitli ev otomasyon görevleri için gerekli sensörler (sıcaklık, nem, hareket vb.) ve aktüatörler (anahtarlar, motorlar vb.).
* Güç Kaynağı: Raspberry Pi ve bağlı cihazlar için stabil bir güç kaynağı.
* Ağ Donanımı: Ethernet kablosu veya Wi-Fi adaptörü (statik IP sağlamak için).

1. **Yazılım Gereksinimleri:**

* İşletim Sistemi: Raspberry Pi için Raspbian (Linux tabanlı).
* Python: Otomasyon framework'ünün geliştirilmesi için.
* PHP-HTML: Kullanıcı arayüzünün oluşturulması için.
* MySQL: Veritabanı yönetim sistemi.
* Apache veya Nginx: Web sunucusu.
* Statik IP Servisi: Uzaktan erişim için.

1. **Güvenlik Gereksinimleri:**

* Veri Şifreleme: Hem veri tabanı hem de iletişim kanallarında güçlü şifreleme mekanizmaları kullanılmalıdır.
* Kullanıcı Kimlik Doğrulama: Sisteme erişim için güvenli kimlik doğrulama yöntemleri uygulanmalıdır.
* Güvenlik Güncellemeleri: İşletim sistemi ve yazılım bileşenleri düzenli olarak güncellenmelidir.

1. **Performans Gereksinimleri:**

* Hızlı Yanıt Süresi: Kullanıcı arayüzü ve otomasyon komutları için düşük gecikmeli yanıt süresi.
* Veritabanı Performansı: MySQL veritabanının optimize edilmesi ve düzenli yedekleme yapılması.

1. **Proje Teslimatları:**

* Sistem Analizi ve Tasarımı: Projenin başlangıcında detaylı bir sistem analizi ve tasarımı belgesi hazırlanacaktır.
* Kod Geliştirme: Python ve PHP-HTML dilleri kullanılarak sistemin geliştirilmesi.
* Veritabanı Tasarımı: MySQL kullanarak veri tabanının oluşturulması ve yapılandırılması.
* Kullanıcı Arayüzü: Kullanıcıların kolayca erişebileceği ve kullanabileceği web tabanlı arayüz.
* Test ve Entegrasyon: Sistemin fonksiyonel ve performans testleri yapılarak hataların giderilmesi.
* Kullanım Kılavuzu: Sistemin kurulumu, yapılandırılması ve kullanımı için detaylı bir doküman.

### Kabul ve Kısıtlar

1. **Kabul kriterleri**

* **Temel Fonksiyonlar**

Sistem entegre edilen cihazların temel fonksiyonlarını yerine getirebilmelidir.

* **Uyumluluk ve Entegrasyon**

Sistem mümkün oldukça çok akıllı ev cihazları ve iot elemanını uyumlu halde tutmalı ve kullanıcı kolay bir şekilde cihazı sisteme entegre edebilmeli.

* **Güvenlik ve Gizlilik**

Sistem kullanıcı verilerini gizli tutmalı ve 3. taraflarla paylaşmamalıdır.

* **Performans ve Hata Yönetimi**

Sistem kesintisiz çalışmalı ve hatalı durumlar için kayıtlar ve hata mesajları oluşturabilmelidir

* **Kullanıcı Deneyimi**

Arayüz kullanıcı dostu ve kullanıcı tarafından modlanabilir halde olmalıdır.

1. **Kısıtlar**

* **Maliyet**

Akıllı ev sistemleri oluşturmak ve bu sistemin otomasyonunu sağlamak oldukça maliyetlidir bu nedenle geliştirilecek sistem oluşturulan bütçe ile sistemin verimliliği doğru orantılı olcaktır.

* **Enerji Kesintileri**

Sistemin sağlıklı çalışabilmesi için sistemi besleyebilecek kesintisiz güç kaynağına ihtiyaç duyulmaktadır. Sistemin sınırsız enerji harcaması verimli olmayacaktır.

* **Entegrasyon Karşıtlığı**

Bazı cihazlar özellikle sistem ile entegre çalışmayı reddedebilir bundan sistem entegrasyon engeli olmayan cihazlarla sınırlıdır.

### UML

1. **Use-Case diyagramı**

metin, diyagram, taslak, çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Sistemimizde yer alan kullanıcılarımızı ve durumları belirleyip incelediğimiz diyagramımızdır.
* Kullanıcı Akıllı et Otomasyon sistemi yöneten ve kontrol eden kişidir. Genellikle ev sahibi veya evde yaşayan aile üyesi olabilir.
* Akıllı cihazlar Akıllı et Otomasyon sistemindeki cihazları kullanarak veri toplayan aktördür.
* Sistem Erişimi, kullanıcı dinamik kontrolcü aracılığıyla ev otomasyon sistemine erişir. Bu süreçte, kullanıcının çeşitli cihazları kontrol edebilmesi için dinamik kontrolcü ile etkileşimde bulunması gerekir.
* Kullanıcı, Sistem erişimi veya Akıllı cihazlar üzerinden kapı kilitlerini açar veya kilitler. Akıllı kilitler, kullanıcının verdiği komutları yerine getirerek evin güvenliğini sağlamaktadır
* Kullanıcı, akıllı lambaları daha önceden kaydettiği ayarlara göre veya çeşitli sensör ve sistem erişim aracılığıyla açar, kapar veya ışık seviyesini ayarlayabilir.
* Kullanıcı, Sistem erişimini kullanarak güvenlik kameralarının görüntülerini izleyebilir, evin iç ve dış güvenliğini kontrol edebilir. Canlı görüntüleri izleme veya kayıtları kontrol etme işlemleri, kullanıcının güvenlik durumunu sürekli olarak gözlemlemesine olanak tanır.
* Kullanıcı, daha önceden kaydettiği ayarlara göre veya sistem erişimi üzerinden evin ısıtma ve soğutma sistemini yönetebilir.
* Dinamik kontrolcü, akıllı depolama biriminde verileri depolar ve yönetir. Tüm cihazlardan gelen verileri toplar, analiz eder ve gerektiğinde bu verilere erişir, böylece sistemin verimli bir şekilde çalışmasını sağlar.

1. **SEQUENCE DİYAGRAMI**

metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Kullanıcı, öncelikle sistem erişimine giriş talebinde bulunur. Sistem erişimi, kullanıcının kimlik bilgilerini doğrulayarak giriş işlemini gerçekleştirir ve kullanıcı bilgilerini kontrol eder veya kaydeder. Bu adımlarla birlikte, kullanıcı sisteme başarılı bir şekilde giriş yapmış olur. Giriş işleminin tamamlanmasının ardından, kullanıcı artık sistem erişimi kısmını geçerek doğrudan kontrolcü ile iletişim kurabilir.
* Işıkların açılması, sıcaklık ayarlaması ve güvenlik kilitlerinin etkinleştirilmesi gibi işlemler doğrudan dinamik kontrolcü üzerinden akıllı cihazlara yönlendirilir. Akıllı cihazlar, bu işlemleri onaylayarak gerçekleştirir ve yapılan işlemlerin durumunu kullanıcıya geri bildirir.
* Kameradan görüntüleri alma işleminde, kullanıcı isteği alındıktan sonra, akıllı depolama sisteminden kaydedilen veya anlık kayıt halinde olan görüntüler çekilerek kullanıcıya aktarılır.

1. **ACTİVİTY DİYAGRAMI**

diyagram, taslak, teknik çizim, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Diyagramın başlangıç noktası, kullanıcının sisteme erişim talebiyle başlar
* Kullanıcı, sistem erişimine giriş yapmak için kimlik bilgilerini girer.
* Sistem erişimi, kullanıcının kimlik bilgilerini doğrular. Eğer kimlik bilgileri doğruysa, kullanıcı sisteme giriş yapar. Eğer kimlik bilgileri yanlışsa, kullanıcıya hata mesajı gösterilir ve tekrar giriş yapması istenir.
* Kullanıcı başarılı bir şekilde giriş yaptığında, arayüz görüntüleme kısmına yönlendirilir. Buradan çeşitli kontrol seçeneklerine erişebilir.
* Kullanıcı, ışıkları açmak veya kapatmak, sıcaklığı ayarlamak, güvenlik kilitlerini etkinleştirmek ve kameradan görüntü almak gibi çeşitli işlemleri gerçekleştirebilir.
* Dinamik kontrolcü, ışıklar, sıcaklık, güvenlik kilitleri ve kamera ile ilgili işlemleri akıllı cihazlara yönlendirir.
* Akıllı cihazlar, bu işlemleri onaylar ve durumlarını kullanıcıya bildirir.
* Kameradan görüntü alma işleminde, istek alındıktan sonra, akıllı depolama sisteminden kaydedilen veya anlık kayıt halinde olan görüntüler çekilir ve kullanıcıya aktarılır.

1. **STATE DİYAGRAMI**

metin, diyagram, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Kullanıcı giriş talebinde bulunur (Start)
* Giriş talebi alınır kimlik doğrulama başarılı olursa kimlik kaydedilir aksi durumda giriş talebi durumuna döneriz.
* Eğer kimlik kayıtlı ise kimlik kaydedildi durumuna direkt geçiş yapabiliriz.
* Kimlik kaydedildi durumundan sistem erişimine geçiş yaparız
* Sistem erişiminden kullanıcı tercihleri ile DKS(Dinamik Kontrol Sistemi), Manuel kullanıcı tercihleri ile akıllı sistem erişimi durumuna, yeni cihaz ekleme ile yeni cihaz ekleme durumuna ve sistem çıkışı isteği ile bitiş durumuna geçer.
* DKS akllı cihaz erişimi ile akıllı cihazlar durumuna, kontrol sistem talepleri ile akıllı sistem erişimi durumuna ve işlem tamamlandı ise bitiş durumuna gidebilir
* Akıllı cihazlar topladıkları verileri akıllı sistem erişimi durumuna veya bitiş durumuna gidebilir.
* Akıllı sistem erişimi kontrol sistem talepleri ve talep bulunamadığı şartlarda DKS durumuna geçer ve kullanıcı tercihleri istediğinde ise sistem erişimi durumuna geçer.
* Yeni cihaz ekleme durumu cihaz bilgileri ile akıllı cihazlar durumuna geçer eğer bilgiler eksik veya hatalı ise cihaz ekleme talebi başarısız oldu hatası ile sistem erişimi durumuna geçiş yapar

1. **CLASS**

metin, makbuz, ekran görüntüsü, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

**1. Kullanıcı Sınıfı**

**Özellikler:**

kullaniciID (int): Kullanıcının benzersiz kimliği.

**Metodlar:**

SistemErisimi(int id): Kullanıcının sisteme erişimini sağlar.

Kullanıcı sınıfı, sistemdeki kullanıcıları temsil eder. Her kullanıcının bir kimliği vardır ve bu kimlik üzerinden sisteme erişim sağlanır.

**2. EvOtomasyonSistemi Sınıfı**

**Özellikler:**

cihazlar (Cihaz[]): Sistemde yer alan cihazların listesi.

senaryo (Senaryo): Sistemde tanımlı senaryolar.

**Metodlar:**

Cihaz cihazEkle(Cihaz c): Sisteme yeni bir cihaz ekler.

Cihaz cihazCikar(Cihaz c): Sistemde mevcut bir cihazı kaldırır.

void senaryoCalistir(Senaryo s): Belirli bir senaryoyu çalıştırır.

EvOtomasyonSistemi sınıfı, sistemdeki tüm cihazları ve senaryoları yönetir. Cihaz ekleme, çıkarma ve senaryo çalıştırma işlemlerini içerir.

**3. Cihaz Sınıfı**

**Özellikler:**

cihazID (String): Cihazın benzersiz kimliği.

cihazAdi (String): Cihazın adı.

cihazDurumu (String): Cihazın mevcut durumu.

Cihaz sınıfı, sistemdeki genel cihazları temsil eder. Her cihazın bir kimliği, adı ve durumu vardır.

**4. Aydınlatma Sınıfı**

**Özellikler:**

parlaklikSeviyesi (int): Aydınlatma cihazının parlaklık seviyesi.

Metodlar:

void parlaklikAyarla(int seviye): Aydınlatmanın parlaklık seviyesini ayarlar.

int parlaklikOgren(): Aydınlatmanın mevcut parlaklık seviyesini öğrenir.

Aydınlatma sınıfı, genel Cihaz sınıfından türetilmiş özel bir cihaz türüdür. Parlaklık seviyesini ayarlama ve öğrenme işlevlerini içerir.

**5. GüvenlikKamerası Sınıfı**

**Metodlar:**

void kayitBaslat(): Güvenlik kamerasında kaydı başlatır.

void kayitDurdur(): Güvenlik kamerasında kaydı durdurur.

GüvenlikKamerası sınıfı, genel Cihaz sınıfından türetilmiş başka bir özel cihaz türüdür. Kayıt başlatma ve durdurma işlevlerini içerir.

**6. Termostat Sınıfı**

**Özellikler:**

sicaklik (double): Termostatın ayarlanmış sıcaklık değeri.

**Metodlar:**

void sicaklikAyarla(double sicaklik): Termostatın sıcaklık değerini ayarlar.

double sicaklikOgren(): Termostatın mevcut sıcaklık değerini öğrenir.

Termostat sınıfı, genel Cihaz sınıfından türetilmiş bir diğer özel cihaz türüdür. Sıcaklık ayarlama ve öğrenme işlevlerini içerir.

**7. Senaryo Sınıfı**

**Özellikler:**

senaryoAdi (String): Senaryonun adı.

komutlar (Komut[]): Senaryoda yer alan komutlar.

**Metodlar:**

String senaryoEkle(String senaryo, Komut k): Senaryoya yeni bir komut ekler.

void senaryoCikar(String senaryo, Komut k): Senaryodan bir komutu çıkarır.

Senaryo sınıfı, belirli durumlarda çalıştırılacak bir dizi komutu içerir. Senaryolar, kullanıcıların belirli koşullar altında otomatik olarak gerçekleştirilmesini istedikleri işlemleri tanımlar.

**8. Komut Sınıfı**

**Özellikler:**

cihaz (Cihaz): Komutun uygulanacağı cihaz.

aksiyon (String): Komutun gerçekleştireceği eylem.

**Metodlar:**

void komutCalistir(Cihaz c, String aksiyon): Belirtilen cihaz üzerinde komutu gerçekleştirir.

Komut sınıfı, belirli bir cihaz üzerinde gerçekleştirilecek işlemleri tanımlar. Bu, cihazın durumunu değiştirmek veya belirli bir eylemi tetiklemek için kullanılır.

**Sınıflar Arası İlişkiler**

* EvOtomasyonSistemi ve Kullanıcı: Bir Kullanıcı, EvOtomasyonSistemi'ne erişebilir (1..\* ilişki). Birden fazla kullanıcı sisteme erişebilir.
* EvOtomasyonSistemi ve Cihaz: EvOtomasyonSistemi, bir veya daha fazla Cihaz içerebilir (1..\* ilişki). Bir sistem, birden fazla cihazı yönetebilir.
* EvOtomasyonSistemi ve Senaryo: EvOtomasyonSistemi, bir veya daha fazla Senaryo içerebilir (1..\* ilişki). Bir sistem, birden fazla senaryoyu yönetebilir.
* Senaryo ve Komut: Bir Senaryo, bir veya daha fazla Komut içerebilir (1..\* ilişki). Bir senaryo, birden fazla komutu içerebilir.
* Komut ve Cihaz: Bir Komut, belirli bir Cihaz üzerinde işlem gerçekleştirir (1..1 ilişki). Her komut, bir cihaza uygulanır.
* Cihaz ve Alt Sınıfları (Aydınlatma, GüvenlikKamerası, Termostat): Cihaz sınıfı, Aydınlatma, GüvenlikKamerası ve Termostat sınıfları için genel bir üst sınıftır (inheritance ilişki). Bu sınıflar, Cihaz sınıfının özelliklerini miras alır ve kendilerine özgü ek özellikler ve metodlar içerir.

### KOD

**YAML**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu YAML yapılandırması, ev otomasyon sisteminde batarya ile ilgili bilgileri kullanıcıya sunmak ve bataryanın durumu, şarj durumu gibi bilgileri görselleştirmek için kullanılır. Değişkenler, bataryanın adını, durumunu, şarj türünü ve batarya seviyelerini belirler ve buna göre kullanıcı arayüzünde görsel geri bildirim sağlar. Renk kodları, batarya seviyesinin durumunu (tehlikeli, uyarı veya iyi) belirtmek için kullanılır.

**SQlite**

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**(komutlar) Tablosu:**

Sistemdeki tüm komutların kaydını tutar ve her komutun türü, verisi, kaynağı ve zaman bilgileri gibi detayları saklar. Bu, sistemdeki komutların izlenebilirliğini ve yönetimini sağlar.

**(sema\_degisikligi) Tablosu:**

Veritabanı şemasında yapılan değişikliklerin kaydını tutar. Şemanın hangi versiyonda olduğunu ve ne zaman değişiklik yapıldığını izlemeye yarar. Bu, şema değişikliklerinin tarihçesini ve yönetimini sağlar.

**(senaryolar) Tablosu:**

Ev otomasyon sistemindeki tüm senaryoları ve bunlara ait bilgileri saklar. Senaryolar, belirli durumlarda otomatik olarak gerçekleştirilecek işlemleri tanımlar ve yönetir. Bu tablo, senaryoların izlenmesini, güncellenmesini ve ilgili komutlarla ilişkilendirilmesini sağlar.

Bu tablolar birlikte çalışarak, ev otomasyon sisteminin işleyişini destekleyen ve verilerin düzgün bir şekilde saklanmasını, yönetilmesini ve izlenmesini sağlayan bir veritabanı altyapısı oluşturur.

## PROJE YÖNETİMİ

### İş- Zaman Çize**lg**esi

**İŞ-ZAMAN ÇİZELGESİ**

ekran görüntüsü, metin, renklilik, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **İP No** | **İş Paketlerinin Adı ve Hedefleri** | **Kim(ler) Tarafından Gerçekleştirileceği** | **Zaman Aralığı (..-.. Ay)** | **Projenin Başarısına Katkısı** |
| 1 | Planlama ve Tasarım | Seyfullah Kurt  Aziz Yapar | 1-2 Ay | %10→%10 |
| D | Parçaların Temini | Seyfullah Kurt | 3-4 Ay | %5→%15 |
| 3 | Yazılım Geliştirme | Seyfullah Kurt  Aziz Yapar, | 4-8 AY | %35→%50 |
| 4 | Entegrasyonların Yapılması | Aziz Yapar | 6-9 Ay | %35→%85 |
| 5 | Sistem Testleri ve Optimizasyon | Seyfullah Kurt | 10-11 Ay | %10→%95 |
| 6 | Sonuçların değerlendirilmesi | Aziz Yapar | 12 Ay | %5→%100 |

### Risk Tablosu

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Olasılık** | **Etki** | **Risk Azaltma Planı** |
| **Entegrasyon Eksikliği veya Uyumsuzluk** | Yüksek | Orta | Sürekli sistem güncellemeleri ve tam entegrasyon için iş ortaklıkları kurulacak. |
| **Sistem Güncellemeleri ile Geriye Dönük Uyumsuzluk** | Düşük | Orta | Sistem için daha kapsayıcı dinamik tasarım oluşturulacak. |
| **Kullanıcılar Arası Yetki Zorbalığı** | Düşük | Yüksek | Sistemdeki her cihaz sadece yetkisi olan kişi tarafından çalışabilecek şekilde tasarlanacak. |
| **Dışardan Sisteme Sızılması** | Orta | Yüksek | Sistem sadece bluetooth cihazla alınabilen key’ler ve admin izni ile kayıt yapan kullanıcıları kabul edecek. |

### Bütçe Talep Çizelgesi

|  |  |  |
| --- | --- | --- |
| **Bütçe Türü** | **Talep Edilen Bütçe Miktarı (TL)** | **Talep Gerekçesi** |
| **Sarf Malzeme** | 3,000 | Yazılım geliştirilirken kullanılacak (CD, USB, kırtasiye malzemeleri, yedek parçalar gibi) |
| **Makine/Teçhizat (Demirbaş)** | 30,000 | Sistemin çalışması için Raspberry Pi cihazları, sensörler, aktüatörler, güç kaynakları gibi |
| **Hizmet Alımı** | 15,000 | Sistemin kurulumu için gerekli altyapı (Elektrik tesisatı, internet altyapısı, danışmanlık hizmetleri) |
| **Ulaşım** | 7,000 | Yapılacak iş hakkında bilgi edinmek için mevcut uygulamaların olduğu yerlere ziyaretler gibi |
| **Personel Giderleri** | 80,000 | Çalışacak kişilerin bir yıllık maaşları |
| **SGK ve İşletme Giderleri** | 37,000 | Çalışanların bir yıllık SGK giderleri ve ofis giderleri (kira, elektrik, su vb.) |
| **Eğitim ve Sertifikasyon** | 10,000 | Proje ekibinin gerekli eğitim ve sertifikasyon masrafları |
| **Toplam** | 182,000 | Projenin başarıyla gerçekleştirilebilmesi için gerekli bütçe |

## EKLER

**EK-1: KAYNAKLAR**

Güğül, G. N., & Sarıtaş, M. (2011). AKILLI EV SİSTEMLERİ VE UYGULAMASI. *Journal of Science and Technology of Dumlupınar University*, (025), 49-60.

İlkbahar, F., Ünal, Ş., Karakaya, A. T., & Eren, B. (2021). Akıllı ev sistemleri üzerine bir model önerisi. *AJIT-e: Academic Journal of Information Technology*, *12*(45), 90-105.

Koçyiğit, Y., & Sine, Ö. (2020). İnternet üzerinden kontrol edilen tam otomasyonlu akıllı ev sistemleri için örnek bir uygulama. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *11*(2), 521-532.