

Akıllı Ev Sistemleri Otomasyon Programı

Melih İyigören 221307034

Kocaeli Üniversitesi Teknoloji Fakültesi

melih_iyigoren@hotmail.com

Abstract—Günümüzde teknolojinin gelişmesiyle birlikte akıllı ev sistemleri, konutlarda güvenliği artırma, enerji tasarrufu sağlama ve yaşam konforunu iyileştirme amacıyla yaygın şekilde kullanılmaktadır. Bu proje kapsamında geliştirilen "Online Akıllı Ev Sistemi Otomasyon Programı", kullanıcıların birden fazla evi için oda, cihaz ve sensör yönetimini dijital ortamda kolayca gerçekleştirmesini amaçlamaktadır. Kullanıcılar sisteme giriş yaparak evlerini ve odalarını tanımlayabilmekte, cihazları uzaktan kontrol edebilmekte ve sensör verilerini anlık olarak takip edebilmektedir. Projede kullanıcı dostu bir arayüz ve yönetim paneli sunulmuş, sistem yöneticisi (admin) için kullanıcı, ev, oda, cihaz ve sensörleri merkezi olarak yönetebileceği ayrı bir panel tasarlanmıştır.

Projenin temel amacı; kullanıcıların farklı lokasyonlardaki evlerini tek bir sistem üzerinden yönetebilecekleri, sade ve işlevsel bir yazılım geliştirmektir. Aynı zamanda veritabanı ilişkileri, güvenli kullanıcı yönetimi ve gerçek zamanlı cihaz/sensör kontrolü gibi temel yazılım mühendisliği bileşenleri başarıyla entegre edilmiştir.

Keywords—Akıllı Ev Otomasyonu, IoT Tabanlı Kontrol, Sensör Verisi İzleme, Kullanıcı Yönetimi, Web Tabanlı Uygulama, PostgreSQL Veritabanı

I. GİRİŞ

Günümüzde akıllı ev sistemleri, enerji verimliliği sağlamak, güvenliği artırmak ve yaşam kalitesini yükseltmek amacıyla yaygın olarak kullanılmaktadır. Nesnelerin İnterneti (IoT) teknolojilerinin gelişmesiyle birlikte, kullanıcıların ev ortamlarını uzaktan izleyip kontrol edebileceği sistemlerin önemi giderek artmaktadır. Bu projede, kullanıcıların sahip oldukları evlerdeki odalara bağlı cihaz ve sensörleri yönetebilecekleri, kullanıcı rolleri doğrultusunda farklı yetkilere sahip olabilecekleri bir Akıllı Ev Otomasyon Sistemi geliştirilmiştir. Proje kapsamında geliştirilen sistemde kullanıcılar, sahip oldukları evlere yeni odalar, cihazlar ve sensörler ekleyebilmekte; cihazların durumunu güncelleyebilmekte ve sensör verilerini görüntüleyebilmektedir. Aynı zamanda sistemde bir yönetici (admin) paneli aracılığıyla tüm kullanıcılar, evler, odalar, cihazlar ve sensörler merkezi olarak yönetilebilmektedir. PostgreSQL tabanlı veritabanı yapısı, verilerin bütünlüğünü ve ilişkisel yapısını sağlayacak şekilde 5N (Beş Normal Form) prensiplerine uygun biçimde tasarlanmıştır. Sistem mimarisi; React tabanlı kullanıcı dostu bir frontend, Node.js/Express.js tabanlı bir backend API ve PostgreSQL veritabanı üzerinde şekillendirilmiştir. Ayrıca, veri güvenliğini ve işlevselliği artırmak amacıyla veritabanı düzeyinde index, view ve trigger yapılarına da yer

verilmiştir. Bu çalışmada geliştirilen sistem, akıllı ev konseptine yönelik işlevsel ve modüler bir yazılım çözümüdür. Raporun ilerleyen bölümlerinde, projenin problem tanımı, araştırma süreci, yazılım mimarisi, veritabanı yapısı ve uygulama detayları sunulacaktır.

II. PROBLEM TANIMI

Günümüzde teknolojik gelişmelerle birlikte bireylerin yaşam alanlarında dijitalleşme oranı artmış, ev otomasyonu sistemleri lüks olmaktan çıkıp bir gereksinim haline gelmiştir. Ancak piyasada bulunan birçok akıllı ev çözümü ya yüksek maliyetlidir ya da kullanıcı dostu arayüzlerden yoksundur. Ayrıca, kişisel ihtiyaçlara göre özelleştirilebilir sistemlerin azlığı, kullanıcıların bu sistemleri etkin biçimde kullanmalarını zorlaştırmaktadır. Bu proje, kullanıcıların farklı evlerini, bu evlere bağlı odaları, odalarda yer alan cihazları ve bu cihazlara bağlı sensörleri yönetebilecekleri, kullanıcı bazlı yetkilendirme sistemine sahip, merkezi bir platform ihtiyacına çözüm üretmeyi amaçlamaktadır. Sistemin yalnızca kullanıcı seviyesinde değil, yönetici seviyesinde de tüm verilerin merkezi olarak izlenip kontrol edilebilmesi gereklidir. Ayrıca, verilerin ilişkisel yapısı korunarak verimli sorguların yapılabilmesi için güçlü bir veritabanı tasarımına ihtiyaç duyulmaktadır. Projede çözülmesi hedeflenen problemler özetle şunlardır:

- Kullanıcıların ev, oda, cihaz ve sensör yönetimini merkezi bir arayüz üzerinden gerçekleştirebilmesi,
- Verilerin güvenli ve ilişkisel bir yapıda saklanabilmesi,
- Kullanıcı ve yönetici rollerine göre yetkilendirme yapılabilmesi,
- Gerçek zamanlı sensör verilerinin izlenebilmesi ve kritik durumların tetiklenebilmesi,
- Veritabanı performansının index, view ve trigger kullanımı ile artırılması.

Bu sorunlara yönelik geliştirilen sistem, hem işlevselliği hem de kullanım kolaylığı ile gerçek hayattaki akıllı ev senaryolarına uyarlanabilir bir yazılım çözümü sunmaktadır.

III. YAPILAN ARAŞTIRMALAR

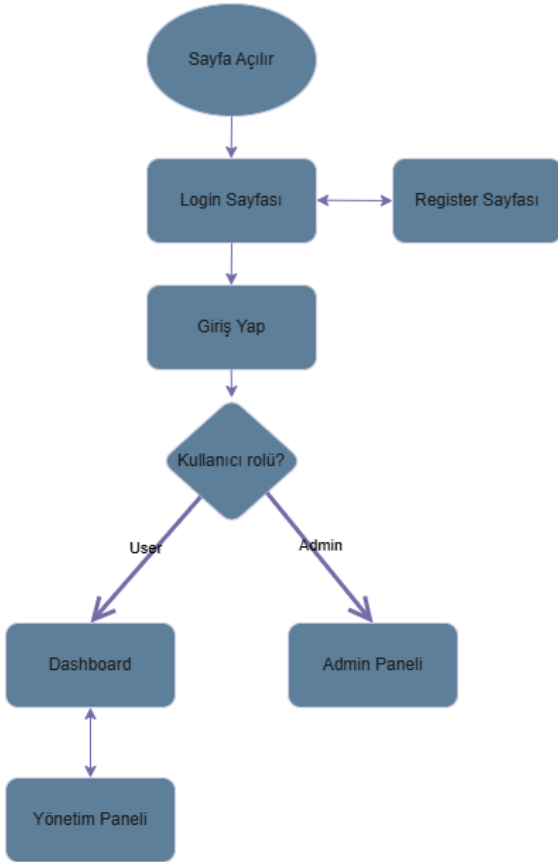
Proje geliştirme süreci boyunca hem yazılım geliştirme hem de veritabanı tasarımı aşamalarında çeşitli zorluklarla karşılaşılmış ve bu zorluklara yönelik araştırmalar yapılarak çözüm yolları üretilmiştir. Araştırmalar genellikle açık kaynak dokümantasyonlar, topluluk forumları, akademik kaynaklar ve teknik bloglar üzerinden yürütülmüştür. İlk olarak, Node.js – PostgreSQL entegrasyonunda pg kütüphanesinin doğru konfigürasyonu ve veritabanı bağlantısının asenkron şekilde yönetilmesi için resmi

PostgreSQL dokümantasyonu ve topluluk kaynaklarından faydalanılmıştır. Veritabanı bağlantısının modüler hale getirilmesi ve proje içinde tekrar kullanılabilir şekilde yapılandırılması, kodun sürdürülebilirliğini artırmıştır. Frontend tarafında, React.js ile kullanıcı arayüzlerinin oluşturulması, component yapısının mantığı, state yönetimi ve router yapısı gibi konular detaylı araştırılmış ve Chakra UI kütüphanesi ile daha kullanıcı dostu, responsive tasarımlar elde edilmiştir. Özellikle dashboard ve yönetim paneli gibi kullanıcı etkileşiminin yoğun olduğu sayfaların tasarımı bu araştırmalar doğrultusunda geliştirilmiştir. Veritabanı tarafında ise normalizasyon kuralları (özellikle 3NF ve üzeri) dikkate alınarak veri tekrarlarının önüne geçilmiş, ilişkisel bütünlük sağlanmıştır. Ayrıca:

- **View** kullanımı ile sık kullanılan JOIN sorgularının sadeleştirilmesi ve performansın artırılması sağlanmıştır.
- **Index**'ler ile sorgu performansı iyileştirilmiş, özellikle filtreleme yapılan sütunlarda hız farkı gözlemlenmiştir.
- **Trigger** mekanizması ile belirli sensör değerleri için otomatik uyarı sistemleri devreye sokulmuştur.

Tüm bu araştırmalar, projenin teknik altyapısını sağlamlaştırmak ve gerçek dünya kullanım senaryolarına uygun bir sistem geliştirmek adına önemli katkı sağlamıştır.

IV. AKIŞ ŞEMASI



Görsel 1 Programın Akış Şeması

V. YAZILIM MIMARISI

A. Frontend

Projenin kullanıcı arayüzü React.js kütüphanesi kullanılarak geliştirilmiştir. React Router ile sayfalar arası geçişler sağlanmış, kullanıcı deneyimini artırmak amacıyla bileşen temelli yapı tercih edilmiştir. Arayüz tasarımı için Chakra UI bileşen kütüphanesi kullanılmış ve bu sayede modern, responsive ve erişilebilir bir arayüz elde edilmiştir. Kullanıcılar login ve register işlemlerinden sonra rollerine göre dashboard veya admin paneline yönlendirilmekte, ayrıca kullanıcılar kendi ev, oda ve cihaz yönetimlerini yine bu arayüz üzerinden gerçekleştirebilmektedir.

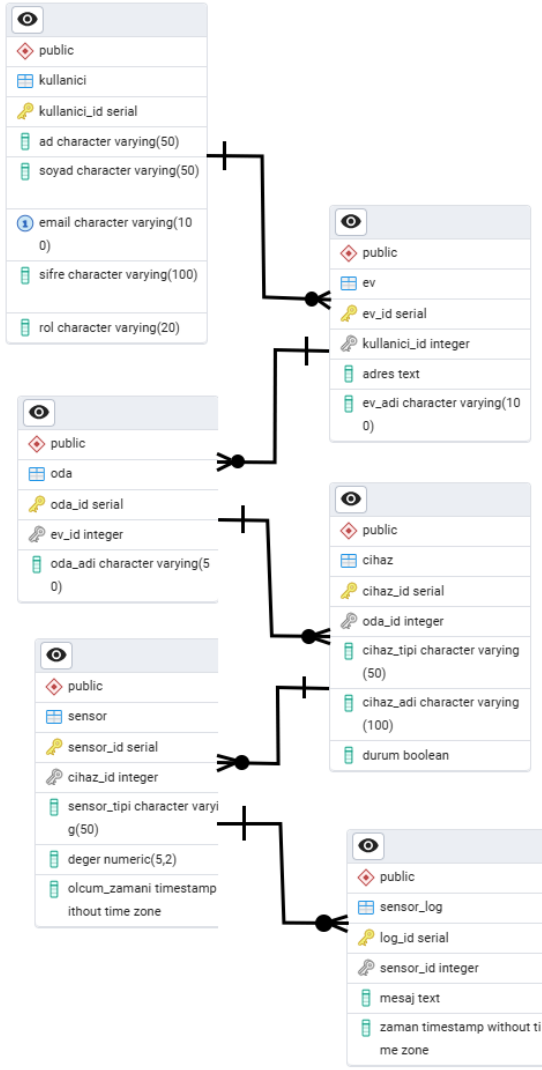
B. Backend

Sunucu tarafı Node.js ortamında Express.js framework'ü kullanılarak hazırlanmıştır. API uç noktaları RESTful prensiplere uygun şekilde yapılandırılmıştır. Kullanıcı kimlik doğrulama işlemleri JSON Web Token (JWT) ile gerçekleştirilmiş, parola güvenliği için bcrypt algoritmasından faydalanılmıştır. Uygulama, kullanıcı rolleri doğrultusunda yetkilendirme sistemine sahiptir ve bu yapı backend'de kontrol edilmektedir. Backend kısmı modüler şekilde yapılandırılmış olup, her tabloya karşılık gelen controller dosyaları üzerinden gerekli işlemler yürütülmektedir.

C. Veritabanı Yapısı

Veri yönetimi PostgreSQL veritabanı sistemi kullanılarak sağlanmıştır. Veritabanında kullanıcı, ev, oda, cihaz ve sensor olmak üzere en az beş temel tablo bulunmaktadır ve bu tablolar birbiriyle ilişkilendirilmiştir. Tasarım sürecinde 5NF (Beşinci Normal Form) normalizasyon kurallarına uyulmuştur. Performansı artırmak amacıyla email, durum ve sensor_tipi gibi sık sorgulanan alanlarda index'ler tanımlanmıştır. Görsel veri erişimini kolaylaştırmak için kullanıcı_cihaz_bilgisi ve kullanıcı_ev_bilgisi gibi view'lar oluşturulmuştur. Ayrıca, sensor tablosuna yapılan veri girişlerinde yüksek değerleri algılamak için sensor_deger_kontrol isimli bir trigger tanımlanmış ve bu durumlarda sensor_log tablosuna otomatik kayıt yapılması sağlanmıştır.

VI. ER DİYAGRAMI



Görsel 2 ER Diyagramı

Bu projede, PostgreSQL, Node.js ve React teknolojileri kullanarak işlevsel ve modüler bir akıllı ev otomasyon sistemi başarıyla geliştirilmiştir. Uygulama, kullanıcıların rollerine göre ev, oda, cihaz ve sensör verilerini yönetebilmesine olanak sağlamaktadır. Cihaz kontrolü, sensör verilerinin takibi ve veri bütünlüğü gibi işlemler normalizasyon kurallarına uygun şekilde gerçekleştirilmiştir. Ayrıca veritabanı yapısında index, view ve trigger gibi önemli kavramlardan etkin biçimde yararlanılmıştır. Proje, ilişkisel veritabanı ile tam entegre çalışan bir full-stack uygulamanın örneğini sunmakta olup, ilerleyen aşamalarda gerçek zamanlı veri takibi, mobil destek veya üçüncü parti sistemlerle entegrasyon gibi özelliklerle geliştirilmeye açıktır.

REFERENCES

- [1] PostgreSQL Documentation. [Online]. Available: <https://www.postgresql.org/docs/>.
- [2] Node.js Documentation. [Online]. Available: <https://nodejs.org/en/docs/>.
- [3] React Documentation – A JavaScript library for building user interfaces. [Online]. Available: <https://react.dev/>.
- [4] Chakra UI – Simple, Modular and Accessible UI Components. [Online]. Available: <https://chakra-ui.com/>.
- [5] Draw.io (diagrams.net) – Flowchart and ER Diagram Tool. [Online]. Available: <https://draw.io/>.
- [6] MDN Web Docs – JavaScript, HTML, CSS Tutorials. [Online]. Available: <https://developer.mozilla.org/>.
- [7] Stack Overflow – Developer Community Forum. [Online]. Available: <https://stackoverflow.com/>.
- [8] W3Schools – Web Development Tutorials. [Online]. Available: <https://www.w3schools.com/>.