

# VERİ TABANI YÖNETİM SİSTEMLERİ PROJESİ

## UÇAK FİRMASI OTOMASYONU

Düzgün Murat Ertik  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
dmuratertik1@gmail.com

Emirhan Güler  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
eguler\_73@hotmail.com

Mehmet Emir Salihoğlu  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
m.emir\_salihoglu@outlook.com

**Özet—** Bu makalede, bir havayolu şirketi için çevrim içi uçak bileti rezervasyonu sistemi tasarımı ve uygulaması sunulmaktadır. Sistem, kullanıcıların uçuş araması yapması, uygun koltukları seçmesi ve biletlerini güvenli bir şekilde satın alması için internet aracılığıyla uçuş rezervasyonu yapmasını sağlar. Sistem; HTML, CSS, JavaScript, Node.js, Express.js ve MySQL teknolojileri bir arada kullanılarak geliştirilmiştir. Sistemin ön ucu, kullanıcıların sitede gezinirken, uçuş seçimi ve koltuk seçimi yaparken ve ödeme yaparken sorunsuz bir deneyim yaşamalarını sağlamak amacıyla oluşturulmuştur. Sistemin arka ucu, tüm uçuşlara ve kullanıcılara ait bilgilerin bulunduğu veri tabanını yönetir ve veri bütünlüğünü sağlar. Genel olarak sistem, kullanıcıların çevrim içi olarak uçak bileti rezervasyonu yapabilmeleri için pratik ve etkili bir yol sunmaktadır.

**Anahtar Kelimeler—** Seyahat Otomasyonu, HTML, CSS, JavaScript, SQL, Node.js, Express.js, MySQL Workbench, ER Diyagramı

### 1. GİRİŞ

Otomasyon, işin insanlar ve makineler arasında paylaştırılması ve iş süreçlerinin daha verimli, hızlı ve hatasız bir şekilde yürütülmesini sağlayan bir süreç olarak tanımlanabilir. Bu süreçte belirli görevler ve iş adımları genellikle teknolojik araçlar veya sistemler tarafından, minimum insan müdahalesiyle gerçekleştirilir ve bu süreç insanların daha stratejik veya karmaşık görevlere odaklanmasına olanak tanır.

Otomasyon, endüstride büyük önem taşıyan bir konudur. Üretimden lojistiğe, güvenlikten veri analizine kadar birçok alanda kullanılan otomasyon sistemleri, iş süreçlerini daha verimli hale getirirken aynı zamanda maliyetleri de düşürmektedir. Bu sayede, karar verme süreçleri iyileştirilir ve verimlilik artırılır. Tüm bu faktörler göz önüne alındığında, otomasyonun endüstrideki önemi ve yaygın kullanımı açıkça görülmektedir. Günümüzde seyahat endüstrisi, teknolojinin gelişmesiyle birlikte büyük bir dönüşüm geçiren alanlardan birisidir. Bu dönüşümün bir parçası olarak, seyahat planlama süreçlerinin dijitalleşmesi ve otomasyonu büyük önem kazanmıştır. Otomasyon, rezervasyon işlemlerinden müşteri hizmetlerine kadar birçok alanda büyük avantajlar sağlamaktadır. Çevrim içi rezervasyon sistemleri, müşterilerin kolayca uçuşları arayıp, seçmelerini ve rezervasyon yapmalarını sağlar, bu da müşteri memnuniyetini artırırken işletmelerin operasyonel verimliliğini artırır. Ayrıca, otomatik check-in sistemleri ve bagaj takip sistemleri gibi otomasyon teknolojileri, seyahat deneyimini daha hızlı ve sorunsuz hale getirirken,

işletmelerin maliyetlerini azaltır. Seyahat endüstrisindeki otomasyon, hem müşterilere hem de işletmelere çeşitli avantajlar sunarak sektördeki rekabeti artırır ve daha verimli bir hizmet sunumunu sağlar.

Bu proje, seyahat endüstrisindeki dijital dönüşümü vurgulamakta ve seyahat planlama sürecinin nasıl iyileştirilebileceğini göstermektedir. Ayrıca, modern teknolojilerin kullanımının seyahat deneyimini nasıl artırabileceğini de göstermektedir. Proje kapsamında, seyahat otomasyonu konsepti temel alınarak, bir havayolu şirketi için çevrim içi bilet satışı yapabilen kapsamlı bir web sitesi ve bu web sitesiyle ilişkili bir veri tabanı oluşturulması amaçlanmıştır. Projenin odak noktası, kullanıcıların seyahat planlarını kolaylaştırmak ve bilet satın alma sürecini daha verimli hale getirmektir.

Web sitesi uçuş arama, rezervasyon ve ödeme işlemleri için kullanıcı dostu bir platform sunar. Kullanıcılar seyahat tarihlerini seçerek uçuşları arayabilir, müsait uçuşları ve uçuşlara ait koltuk bilgilerini görüntüleyebilir, rezervasyonlarını tamamlayabilirler. Ödeme işlemi tamamlandıktan sonra rezervasyon onayı müşteriye iletilir. Veri tabanı tarafında, MySQL Workbench kullanılarak uçuşlar, müşteri bilgileri, rezervasyonlar ve ödeme bilgileri gibi veriler saklandı ve bu verilerin yönetimi sağlanmıştır. Bu veri tabanı, web sitesinin tüm işlevlerini desteklemek için gereken verileri sağlamakta ve saklamaktadır.

### 2. KULLANILAN TEKNOLOJİLER VE YAPILAN ARAŞTIRMALAR

#### 2.1. Web Sitesi

Web sitesinde HTML, CSS, JavaScript, Bootstrap, jQuery, Node.js ve Express.js gibi teknolojiler kullanılmıştır. HTML ve CSS, sayfanın yapısını ve stilini belirlemek için, JavaScript ve jQuery ise kullanıcı etkileşimleri ve veri yönetimi için kullanılmıştır. Bootstrap, responsive özellik ve kullanıcı arayüzü bileşenleri için kullanılan bir CSS ve JS kütüphanesi olarak kullanılmaktadır. Sayfanın görünümünü düzenlemede Font Awesome ise ikonlar için kullanılmıştır. Ayrıca, tarih ve saat işlemleri için moment.js ve tempusdominus-bootstrap-4 kütüphaneleri de kullanılmıştır. Node.js ve Express.js ise sunucu tarafı işlemlerini yönetmek üzere kullanılmıştır.

Kodun yazımı ve kullanılan teknolojiler hakkında bilgi edinme esnasında çeşitli kaynaklardan yararlanılmıştır:

- YouTube platformunda konular hakkında oluşturulmuş videolar ve oynatma listeleri.[1][2]

- Web sitesi görünümü için örnekler ve bu örnekler ait kodlar barındıran web siteleri.[3]
- Programlama sürecinde karşılaşılan sorunlara ait çözümlerin bulunduğu siteler.[4]
- Kullanılan dillere ait bilgilerin bulunduğu siteler.[5][6]
- Kullanıcıların sorularını cevaplamak ve çeşitli konularda yardımcı olmak için eğitilmiş yapay zeka tabanlı dil modelleri.[7][8]

Web sayfalarının yapısal içeriğini tanımlamak için HTML (HyperText Markup Language), kullanılan bir işaretleme dilidir. CSS (Cascading Style Sheets), HTML elemanlarının görsel sunumunu belirlemek için kullanılır. JavaScript, tarayıcıda çalışan ve web sayfalarını dinamiklik kazandıran bir programlama dilidir. Bu üç dile hakim olmak adına YouTube üzerinden çeşitli videolar izlendi, farklı yazılım içerikli sitelerden, diller hakkında bilgiler ve örnekler edinildi. Kodun yazım aşamasında yapay zekalardan yardım alındı.

Bootstrap, web sitesi tasarımını kolaylaştırmak ve hızlandırmak için kullanılan bir CSS ve JavaScript kütüphanesidir. jQuery, JavaScript kütüphanesidir ve HTML belgelerindeki olayları işlemek, animasyonlar oluşturmak, AJAX ile veri alışverişi yapmak gibi işlemleri kolaylaştırır. Bu iki kütüphaneye hakim olmak adına farklı yazılım içerikli sitelerden bilgiler edinildi. Kodun yazım aşamasında yapay zekalardan yardım alındı.

Node.js, sunucu tarafında JavaScript çalıştırmak için kullanılan bir ortamdır. Express.js, Node.js tabanlı web uygulamaları oluşturmak için kullanılan bir web uygulama çerçevesidir. Node.js, JavaScript tabanlı olması ve hem sunucu tarafında hem de istemci tarafında aynı dilin kullanılabilmesi nedeniyle büyük bir avantaj sağlar. Bu durum, kodun daha tutarlı ve daha kolay anlaşılabilir olmasını sağlar. Express.js'in hızlı, esnek ve kolay kullanımı, web uygulamaları ve API'ler geliştirirken tercih edilen bir çerçeve haline gelmesini sağlar. Bu nedenle, Express.js kullanımına karar verilmiştir. Node.js ve Express.js kullanımına karar verilirken web sitelerinden edinilen bilgi ve projenin yapımından sorumlu grup üyelerinin proje yapımında edindikleri tecrübeler göz önüne alındı. Kodun yazım aşamasında yapay zekalardan yardım alındı.

Moment.js, JavaScript tarih ve saat nesnelerini işlemek ve biçimlendirmek için kullanılan bir kütüphanedir. Tempus Dominus Bootstrap 4, Bootstrap ile entegre edilen bir tarih ve saat seçici kütüphanesidir, kullanıcıların tarih ve saatleri kolayca seçmelerine olanak tanır. Font Awesome ise web sitelerinde kullanılan simgeler için vektör tabanlı bir ikon kütüphanesidir. Kaynaklardan edinilen bilgiler ve yönergeler doğrultusunda bu teknolojilerin kullanımına karar verilmiştir.

## 2.2. Veri Tabanı

Veri tabanı oluşturmak için MySQL Workbench kullanıldı. Projenin ait olduğu derslerde bu uygulamanın kullanımının aktarılması sebebiyle MySQL Workbench kullanımı uygun görüldü.

Başlangıçta 8 tabloyla başlandı. Ödeme işlemleri genellikle harici bir servis veya ödeme işlemi sistemi tarafından yönetildiği için, ödeme bilgilerinin veritabanında tutulması gerekli görülmedi. Bu doğrultuda payment tablosu

silindi. Silinen tablo haricinde oluşturduğumuz tablolar şunlardır: airports, booking, customers, citizenship, flights, invoices ve seat. Bu tablolar, standart bir havayolu rezervasyon sistemi için tipik olan veri yapılarını içerir.

Veri tabanı oluşturulurken karşılaşılan sıkıntıların çözümü ve web sitesi ile veri tabanı arasındaki iletişimi sağlayan bazı karmaşık sorguların barındırdığı hataların tespiti için yapay zekalardan yardım alındı. Ayrıca, her bir sorgu koda eklenmeden önce MySQL Workbench üzerinde denendi. Grup üyelerinin aynı veri tabanı üzerinde çalışabilmesi adına veri tabanını paylaşabilmek için gerekli işlemlere ait bilgiler YouTube videoları tarafından edinildi.[7]

## 3. SİSTEM TASARIMI

Web sitesi, ana sayfasında uçuş arama formu sunar. Bu formda kullanıcılar kalkış ve varış havalimanlarını, tarihi ve yolcu sayısını seçebilir. Ardından, kullanıcıların yolcu bilgilerini girdiği bir form sayfası gelir. Daha sonra, uçuşların listelendiği bir sayfa açılır ve kullanıcılar seçtikleri uçuşu rezerve edebilir. Kullanıcılar rezervasyon yaparken, koltuk seçim sayfasına yönlendirilir ve burada koltuk seçimlerini yapabilirler. Son olarak, kullanıcıların ödeme yapabileceği bir sayfa gelir ve işlem tamamlandığında başarı mesajı görüntülenir. Web sitesi var olan uçuş firmalarının web siteleri örnek alınarak hazırlanmıştır.

### 3.1. Veri Tabanı Tasarımı

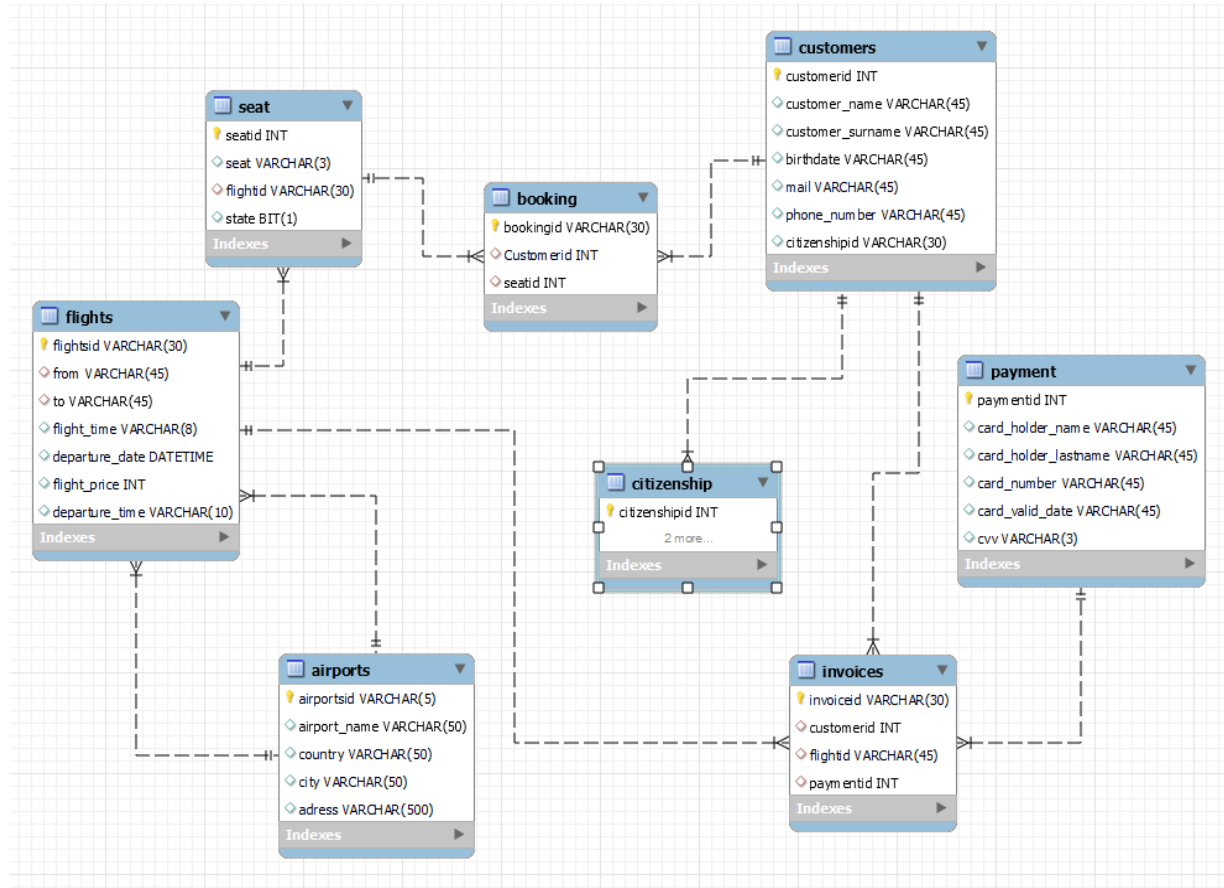
Veri tabanı başta 8 tablo olarak tasarlandı. Ödeme işlemleri genellikle harici bir servis veya ödeme işlemi sistemi tarafından yönetildiği için, ödeme bilgilerinin veritabanında tutulması gerekli görülmedi. Bu doğrultuda payment tablosu silindi. Bununla birlikte ek servis hizmetlerinin, proje yapımı esnasında projeye ait önemli kısımlara odaklanılmasında güçlük çıkaracağı düşünüldüğü için invoices tablosu da silindi. Tablolar silinmeden önceki veri tabanına ait ER Diyagramı (Şekil 1). Veri tabanının son halini gösteren ER Diyagramı (Şekil 2).

Veri tabanı 6 adet tablodan oluşmaktadır:

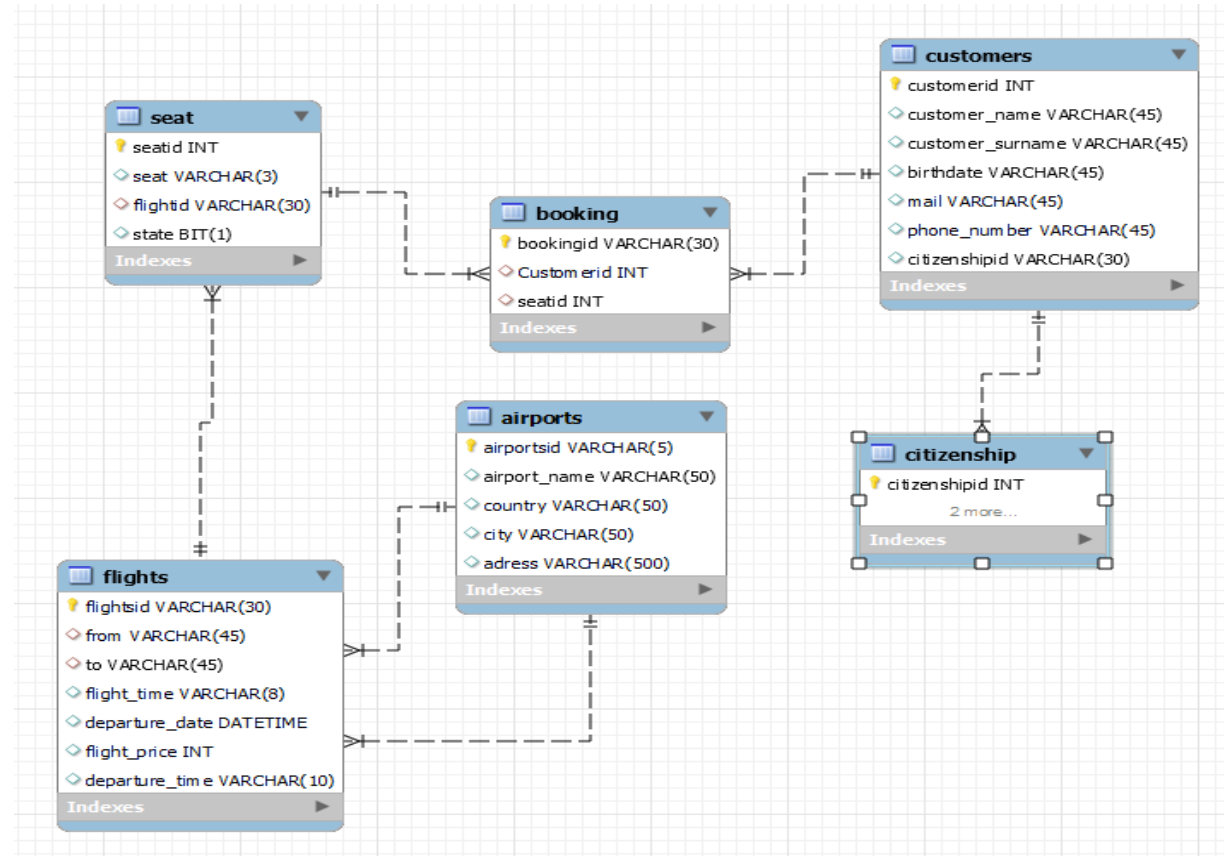
1. airports: airports tablosu, havaalanlarına ilişkin bilgileri içerir. Her havaalanı için benzersiz bir kimlik numarası olan airportsid sütunu bulunur. airport\_name sütunu, havaalanlarının isimlerini içerirken, country ve city sütunları, havaalanlarının bulunduğu ülke ve şehir bilgilerini tutar. Son olarak, adress sütunu, havaalanlarının tam adreslerini içerir. Bu tablo, havaalanları hakkında temel bilgileri depolamak için kullanılır ve uçuş rezervasyon sistemlerinde önemli bir rol oynar.
2. booking: booking tablosu, uçuş rezervasyonlarıyla ilgili bilgileri depolar. Her rezervasyon için benzersiz bir kimlik numarası olan bookingid sütunu bulunur. Customerid sütunu, rezervasyonu yapan müşterinin kimliğini belirtir. Customerid sütunu, customers tablosundaki müşteri bilgileriyle ilişkilidir. Bu sütun, müşterinin kimliğini belirtir ve customers tablosundaki ilgili müşteri kaydına bir referans içerir. Son olarak, seatid sütunu, rezervasyonun hangi koltuk veya koltukları

içerdiğini gösterir. Customerid sütununa benzer şekilde, seatid sütunu, seat tablosundaki koltuk bilgileriyle ilişkilidir. Her seatid değeri, seat tablosundaki ilgili koltuk kaydına bir referans sağlar. Bu ilişkiler sayesinde, her rezervasyonun ilgili müşteri ve koltuk bilgileriyle ilişkilendirilmesi sağlanır. Bu tablo, uçuşların rezervasyon ve koltuk atama süreçlerini yönetmek için kullanılır.

3. customers: customers tablosu, müşteri bilgilerini depolamak için kullanılan bir tablodur. Her müşteri için benzersiz bir kimlik numarası olan customerid sütunu bulunur. customer\_name ve customer\_surname sütunları, müşterinin adını ve soyadını içerir. birthdate sütunu, müşterinin doğum tarihini temsil eder. mail sütunu, müşterinin e-posta adresini içerir. phone\_number sütunu, müşterinin telefon numarasını içerir. Son olarak, citizenshipid sütunu, müşterinin vatandaşlık veya kimlik numarasını temsil eder. Bu sütun, müşterinin vatandaşlık bilgilerini içeren citizenship tablosuyla ilişkilidir ve müşterinin kimlik bilgilerine erişim sağlar. Bu tablo, uçuş rezervasyonları sırasında müşteri bilgilerini kaydetmek ve yönetmek için kullanılır.
4. citizenship: citizenship tablosu, müşterilerin vatandaşlık veya kimlik bilgilerini depolamak için kullanılır. Her bir vatandaşlık veya kimlik numarası için benzersiz bir kimlik numarası olan citizenshipid sütunu bulunur. Bu sütun, müşterilerin kimlik bilgilerini benzersiz bir şekilde tanımlar. citizen sütunu, müşterinin vatandaşlık durumunu belirtir. Bu sütun, müşterinin hangi ülkenin vatandaşı olduğunu veya hangi kimlik belgesine sahip olduğunu gösterir. Müşterinin vatandaşlık veya kimlik bilgilerini daha spesifik şekilde tanımlayan bir sütundur. Bu tablo, müşterilerin vatandaşlık veya kimlik bilgilerini kaydetmek ve yönetmek için kullanılır.
5. flights: flights tablosu, uçuşlarla ilgili bilgileri depolar. Her uçuş için benzersiz bir kimlik numarası olan flightsid sütunu bulunur. from ve to sütunları, uçuşun kalkış ve varış noktalarını belirtir. flight\_time sütunu, uçuşun toplam uçuş süresini gösterir. departure\_date ve departure\_time sütunları, uçuşun kalkış tarihini ve saati belirtir. flight\_price sütunu, uçuşun fiyatını gösterir. Bu tablo, uçuşların bilgilerini depolayarak rezervasyon işlemlerinde kullanılır. Uçuşlar arasında bağlantılar kurularak müşterilere uçuş seçenekleri sunulabilir.
6. seat: seat tablosu, uçak koltuklarının bilgilerini depolar. Her koltuk için benzersiz bir kimlik numarası sağlayan seatid sütunu bulunur. Seat sütunu, koltuğun numarasını veya adını içerir (örneğin, "1A" veya "2B"). flightid sütunu, koltuğun hangi uçuşa ait olduğunu belirtir. Bu sütun, uçuş bilgilerini içeren flights tablosuyla ilişkilidir ve uçuş bilgilerine erişim sağlar. State sütunu, koltuğun durumunu belirler ve genellikle "0" (uygun) veya "1" (uygun değil) gibi değerler alır. Bu tablo, uçak koltuklarının durumunu takip etmek ve rezervasyon yaparken uygun koltukları göstermek için kullanılır.



Şekil 1. Tablolar silinmeden önceki ER Diyagramı.



Şekil 2. Son ER Diyagramı.

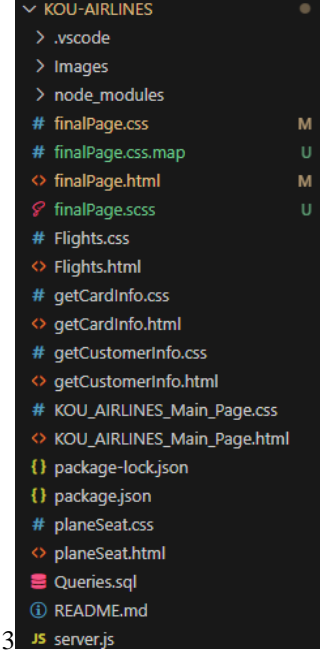
### 3.2. Web Sitesi Tasarımı

Web sitesi toplamda 6 sayfadan olacak şekilde tasarlanmıştır. 6 sayfa ve o sayfalara ait dosyalar (Şekil 3). Sayfalar ve görevleri şu şekildedir:

1. Ana Sayfa: Ana bölümde kullanıcılar, uçuş yapmak istedikleri kalkış ve varış noktalarını seçebiliyor, seyahat tarihini belirleyebiliyor ve kaç kişilik bir rezervasyon yapacaklarını belirtebiliyorlar. Ayrıca, "Search" butonuna tıklayarak arama yapabiliyorlar. Sayfa, Bootstrap ve Font Awesome gibi kütüphanelerden yararlanarak kullanıcı arayüzünü oluşturuyor ve düzenliyor. jQuery kullanılarak sayfa üzerindeki etkileşimler yönetiliyor. Kullanıcıların seçtiği bilgiler (kalkış ve varış noktaları, tarih, müşteri sayısı) JavaScript yardımıyla kontrol edilerek, gerekli alanların doldurulmadığı durumlarda kullanıcıya bir uyarı mesajı gösteriliyor. Son olarak, kullanıcıların yaptığı seçimler localStorage üzerinde saklanarak, sonraki sayfalarda kullanılmak üzere saklanıyor ve sunucudan uçuş verileri çekilerek kullanıcıya sunuluyor. Şekil 3'te "KOU\_AIRLINES\_Main\_Page.html" şeklinde görülebilir.
2. Uçuşlar Sayfası: Ana sayfadaki search butonuna tıklandıktan sonra, veritabanında seçilen bilgileri içeren uçuşlar farklı uçuş kartlarında, kalkış ve varış havaalanları, kalkış ve varış şehirleri, kalkış tarihi ve saati, uçuş süresi ve fiyat bilgileri bulunacak şekilde gösteriliyor. Kullanıcılar, "Book Now" butonuna tıklayarak seçtikleri uçuşu rezerve edebiliyorlar. "Book Now" butonuna tıkladığında, kullanıcının seçtiği uçuşun kalkış saati localStorage'a kaydediliyor ve kullanıcı, müşteri bilgilerini girmek için bir sonraki sayfaya yönlendiriliyor. Sayfa, Bootstrap ve Font Awesome gibi kütüphanelerden yararlanarak kullanıcı arayüzünü oluşturuyor ve düzenliyor. jQuery kullanılarak sayfa üzerindeki etkileşimler yönetiliyor. localStorage kullanılarak kullanıcının seçtiği uçuş bilgileri saklanıyor ve rezervasyon işlemi için bir sonraki adıma geçiş sağlanıyor. Şekil 3'te "Flights.html" şeklinde görülebilir.
3. Müşteri Bilgisi Sayfası: Sayfa, kullanıcıların her bir yolu için ad, soyad, e-posta, telefon numarası, doğum tarihi ve vatandaşlık bilgilerini girebileceği bir arayüz sunuyor. Kullanıcılar, her bir yolcu için ayrı bir form oluşturabiliyorlar. Her formda, yolcunun adı, soyadı, e-posta adresi, telefon numarası, doğum tarihi ve vatandaşlık bilgileri için alanlar bulunuyor. Formların altında, "Next" butonu bulunuyor. Kullanıcılar formları doldurduktan sonra "Next" butonuna tıkladıklarında, sayfa formdaki verilerin geçerli olup olmadığını kontrol ediyor. Eğer

formdaki tüm alanlar doğru şekilde doldurulmuşsa, her bir yolcunun bilgilerini localStorage'a kaydediyor ve kullanıcıyı bir sonraki adıma yönlendiriyor. Eğer formda eksik veya hatalı alanlar varsa, kullanıcıya uyarı vererek eksik alanları doldurması gerektiğini belirtiyor. Şekil 3'te "getCustomerInfo.html" şeklinde görülebilir.

4. Koltuklar Sayfası: Bu sayfada kullanıcılar seçmiş oldukları uçuşa ait koltukların uygunluğunu görüntüleyebilir, uygun koltuklardan müşteri sayısı kadar seçebilir ve seçilen koltuklar localStorage'da saklanır. Ayrıca, kullanıcının seçtiği koltuk sayısına göre bir uyarı verilir ve yeterli koltuk seçilmişse bir sonraki sayfaya yönlendirilir. Şekil 3'te "planeSeat.html" şeklinde görülebilir.
5. Ödeme sayfası: Bu sayfa, kullanıcıların kart bilgilerini girmelerini ve ödeme yapmalarını sağlayan bir form içerir. Form, kart sahibinin adı, kart numarası, son kullanma tarihi ve CVV gibi bilgileri içerir. Formun altında "Pay" adında bir buton bulunur. Bu butona tıklanarak, önce formun doğruluğu kontrol edilir ve eğer form doğrulanırsa, kullanıcının seçtiği koltuk bilgileriyle birlikte sunucuya gönderilir. Sunucudan dönen cevaba göre, ödeme işlemi tamamlanır veya kullanıcıya hata mesajı gösterilir. Şekil 3'te "getCardInfo.html" şeklinde görülebilir.
6. Son Sayfa: Bu sayfa, ödeme işleminin başarıyla tamamlandığını kullanıcıya bildirir. Sayfada, ödeme işlemi tamamlandıktan sonra gösterilecek bir başarı bildirimi bulunmaktadır. Bildirimde, ödeme işlemi başarılı bir şekilde tamamlandığı ve biletin rezerve edildiği bilgisi verilmektedir. Ayrıca, kullanıcıya havalimanında belirtilen zamanda olması gerektiği hatırlatılmakta ve teşekkür edilmektedir. Kullanıcıya iyi bir gün dilerken, uçuş detaylarıyla ilgili soruları olması durumunda iletişime geçmeleri istenmektedir. Şekil 3'te "finalPage.html" şeklinde görülebilir.

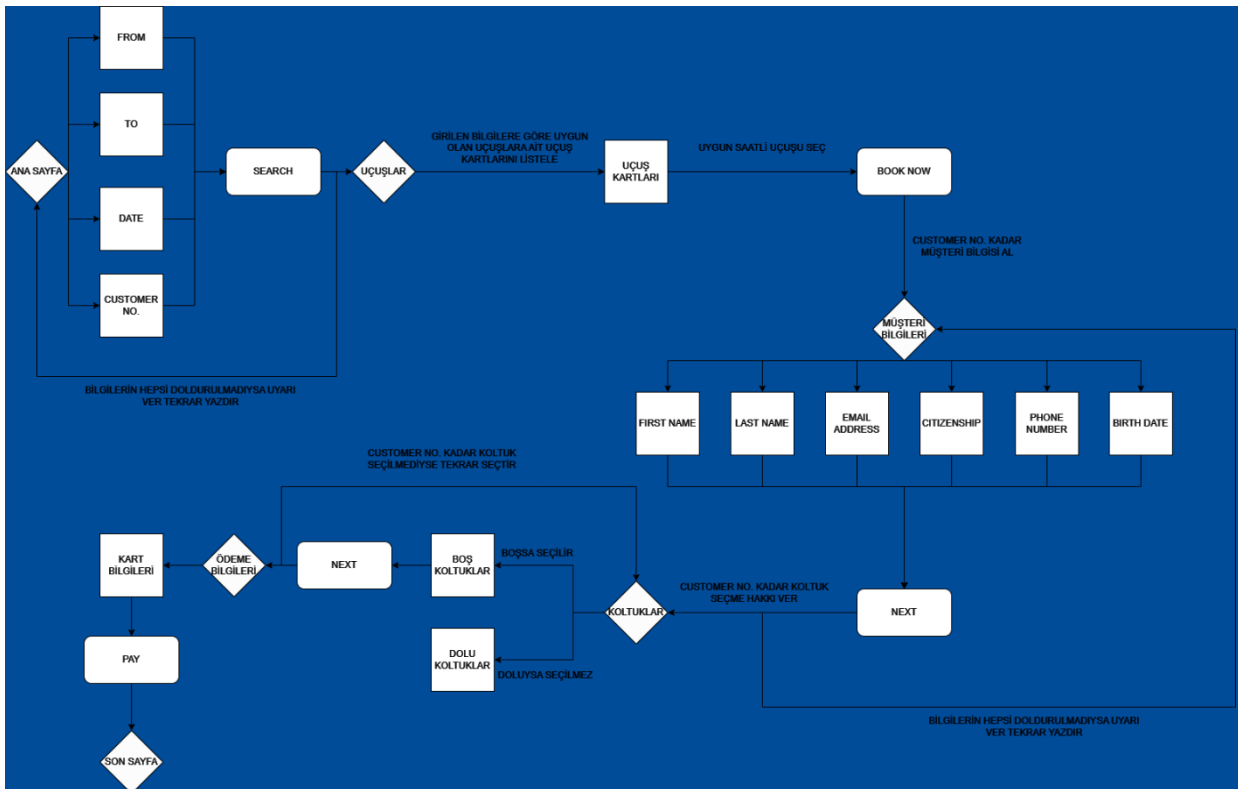


Şekil 3. Projeye ait dosyalar.

#### 4. AKIŞ ŞEMALARI



Şekil 4. Web sitesinin kullanıcıya yönelik sayfa düzenine ait akış şeması.



Şekil 5. Web sitesinin akış şeması.

## REFERANSLAR

1. [https://www.youtube.com/watch?v=G3e-cpL7ofc&ab\\_channel=SuperSimpleDev](https://www.youtube.com/watch?v=G3e-cpL7ofc&ab_channel=SuperSimpleDev)
2. [https://www.youtube.com/playlist?list=PLURN6mxdcwL\\_D8H1iki2YComp-lNyNAdbz](https://www.youtube.com/playlist?list=PLURN6mxdcwL_D8H1iki2YComp-lNyNAdbz)
3. <https://codepen.io>
4. <https://stackoverflow.com>
5. <https://www.w3schools.com>
6. <https://developer.mozilla.org>
7. <https://chatgpt.com>
8. <https://gemini.google.com>