|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 19/07/2022** | | **KAPSAMI: CRUD İşlemleri** |
|  | |  |
| * ASP .NET CORE’da kullanılan MVC yapısını anlayabilmem için kullanıcı eklenebilen, kullanıcının aktiflik pasiflik durumu güncellenebilen ve kullanıcı silinebilen bir sayfa tasarlanmam istendi. Yani kısacası CRUD işlemleri uygulamam istendi. * Projeye başlangıç için Visual Studio’da boş bir MVC projesi oluşturuldu. Bu projenin içine Application, Data, Domain ve Web tabakaları eklendi. * Domain tabakasının içine User class’ı (modeli) eklendi. User modelinin property’leri Id, Username, Password, Email, AccountStatus ve Created olarak belirlendi. User class’ının içinde yeni eklenen kullanıcının aktif olduğunu belirtmek için Domain tabakasının içine Status isimli yeni bir class eklendi ve User class’ının içerisinde AccountStatus property’sinde Status class’ı çağrıldı.   **User Class** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 21/07/2022** | | **KAPSAMI: CRUD İşlemleri (Create User)** |
|  | |  |
| * Database’e veri kaydetmek için UnitOfWork class’ı oluşturuldu.   **UnitOfWork Class** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
| * Web katmanının içinde UserController class’ı oluşturuldu. İçerisine Index, Create(HttpGet), Create(HttpPost) metotları eklendi. Index metodu database’deki kullanıcı verilerini çekip websitesinde bastırmak için kullanıcı verilerini view’e (cshtml dosyasına) gönderdi. Create(HttpGet) metodu kullanıcı verilerini almak için, Create(HttpPost) metodu ise girilen kullanıcı verilerini database’e kaydetmek için kullanıldı. Kullanıcı eklendiğinde \_logger instance’ını kullanarak mylog.txt dosyasına “Yeni Kullanıcı Eklendi” mesajı eklendi. \_logger instance’ını oluşturabilmek için web katmanına Microsoft.Extensions.Logging ve Serilog.Extensions.Logging.File package’ları eklendi.   **Metotlar** | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |
| **Website’de CRUD İşlemlerinin Yapıldığı Sayfa**     * Şu ana kadar hiçbir kullanıcı eklenmediği için tablo boş geliyor. Eğer Add New User butonuna tıklarsak kullanıcı bilgilerinin girildiği sayfa gelir. Örnek olması açısından 2 tane kullanıcı eklendi.        * Kullanıcı bilgilerini girip Kaydet butonuna tıklarsa kullanıcı bilgileri database’e kaydedildi. Daha sonra kullanıcı bilgilerinin görüldüğü sayfa gelir. Buradaki kullanıcı bilgileri database’den çekildi. Ayrıca, kullanıcı başarılı bir şekilde eklenince mylog.txt dosyasına “Yeni Kullanıcı Eklendi” mesajı bastırıldı. | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **YAPILAN İŞİN;** | | | | | | | | | | |
| **TARİHİ: 22/07/2022** | | | | | | | | **KAPSAMI: CRUD İşlemleri (Edit)** | | |
|  | | | | | | | |  | | |
| * UserController class’ının içerisine Edit(HttpGet), Edit(HttpPost) metotları eklendi. Edit(HttpGet) metodu yeni kullanıcı verilerini almak için view’e (cshtml dosyasına) gitti. Edit(HttpPost) metodu ise girilen yeni kullanıcı verilerini database’e kaydetmek için kullanıldı. Kullanıcı bilgileri güncellendiğinde \_logger instance’ını kullanarak mylog.txt dosyasına hangi kullanıcının bilgilerinin değiştirildiği mesajı eklendi.   **Metotlar**    **WebSite’de CRUD İşlemlerinin Yapıldığı Sayfa**     * Eğer Düzenle butonuna tıklarsak kullanıcı bilgilerinin güncellenebileceği sayfa gelir. Örnek olması açısından, Ali kullanıcısı güncellenmek istendi. | | | | | | | | | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | | | | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | | | | | | |
|  | | | |  | | | | | | |
| **YAPILAN İŞİN;** | | | | | | | | | |
| **TARİHİ: 25/07/2022** | | | | | **KAPSAMI: CRUD İşlemleri (Delete)** | | | | |
|  | | | | |  | | | | |
| * UserController class’ının içerisine Delete(HttpGet), DeleteConfirmed(HttpPost) metotları eklendi. Delete(HttpGet) metodu seçilen kullanıcının silinmesinin onaylanması gerektiği için view’e (cshtml dosyasına) gitti. DeleteConfirmed(HttpPost) metodu ise silinmek istenen kullanıcının statüsünü ‘deleted’ yaptı ve bu kullanıcının verilerini database’e kaydetmek için kullanıldı. Silinmek istenen kullanıcı bilgileri database’e kaydedildikten sonra \_logger instance’ını kullanarak mylog.txt dosyasına hangi kullanıcının silineceği mesajı eklendi.   **Metotlar**    **WebSite’de CRUD İşlemlerinin Yapıldığı Sayfa** | | | | | | | | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | | | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | | | | | | |
|  | | |  | | | | | | |
| **YAPILAN İŞİN;** | | | | | | | | |
| **TARİHİ: 26/07/2022** | | | | | | **KAPSAMI: CRUD İşlemleri (Change Status)** | | |
|  | | | | | |  | | |
| * UserController class’ının içerisine ChangeStatus(HttpGet) metodu eklendi. ChangeStatus(HttpGet) metodu seçilen kullanıcının statüsünü değiştirdi. Eğer kullanıcı aktifse ve Statü Değiştir butonuna tıklarsa pasif oldu. Eğer kullanıcı pasifse ve Statü Değiştir butonuna tıklarsa aktif oldu. Ardından, \_logger instance’ını kullanarak mylog.txt dosyasına hangi kullanıcının statüsünün değiştirileceği mesajı eklendi. En son ise, bu kullanıcının verilerini database’e kaydetmek için kullanıldı.   **Change Status Metodu**    **WebSite’de CRUD İşlemlerinin Yapıldığı Sayfa**     * Eğer Statü Değiştir butonuna tıklarsak aktif olan mehmet kullanıcısı artık pasif duruma geldi. | | | | | | | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | | | | | | |
|  | |  | | | | | | |
| **YAPILAN İŞİN;** | | | | | | | | |
| **TARİHİ: 27/07/2022** | | | | | | | **KAPSAMI: Postman’de API Sorguları** | |
|  | | | | | | |  | |
| * Bugün projenin geri kalanı için gerekli olan API konusuna çalışıldı. * Şirketin, otel rezervasyon sistemi için API’ye gönderdiği request metotları fork edildi. * Şirketin otel verilerini sakladığı servis adresi alındı. * Bu servis adresine Postman üzerinden otel rezervasyonu oluşturmak için çeşitli request’ler sırasıyla gönderildi. Bu request’ler header ve body yapısından oluşturuldu. * Bu request’lerden ilki olan Login request’inin body’sinde yer alan variable’lar (agency,user,password) ve servis adresini saklamak için environments kısmında 4 tane environment variable oluşturuldu.     **Request’lerin Body Kısımları**  **Login Request**     * Kullanıcı bilgilerini kullanıp response olarak token almak için bu yöntem   kullanıldı. Buradan alınan token, bundan sonra uygulanacak request’lerde  kullanıldı.  **GetArrivalAutoComplete Request** | | | | | | | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | | | | | | | |
|  |  | | | | | | | |

|  |  |
| --- | --- |
| **PriceSearch Request**       * Otel fiyatlarını almak için PriceSearch request’i kullanıldı. | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |
| --- | --- |
| **GetProductInfo**     * Otel medya bilgilerini fiyatsız almak için GetProductInfo request’i   kullanıldı.  **GetOffers**       * Seçili otel tekliflerini almak için GetOffers request’i kullanıldı.   **BeginTransaction**     * Rezervasyon işlemini başlatmak için BeginTransaction yöntemi kullanıldı. | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 28/07/2022** | | **KAPSAMI: Websitesinde Login Kısmı** |
|  | |  |
| * Postman’de uygulanan Login request’i için interface (ILoginUserService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (LoginUserService.cs) da oluşturuldu. Bu class içerisinde Login isminde bir metot oluşturuldu.   **ILoginUserService.cs**    **LoginUserService.cs İçindeki Login Metodu**     * Implement edilen Login metodunda giriş yapan kullanıcının API üzerinde olup olmadığı kontrol edildi. Login metodunda API’ya atılan request’i modellemek için UserInfo isminde bir class oluşturuldu. Ayrıca, Login metodunda API’ye atılan request’in response’unu modellemek için LoginResponse isminde bir class oluşturuldu. | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
| **UserInfo.cs**    **LoginResponse.cs**     * HomeController içine Index metodu ve LoginUser metotları eklendi. Index metodu, web sitesinin giriş sayfasında bulunacak login kısmını oluşturmak için view’e (Index.cshtml dosyasına) gitti. Kullanıcı giriş yaparken submit butonuna bastığı an ise kod içinde cshtml dosyasından HomeController içindeki LoginUser metoduna gidildi. Login User metodunda LoginUserService içindeki Login metodu çağırıldı. Eğer Login metodu null dönerse yanlış kullanıcı bilgileriyle giriş yapıldı demektir ve bu durumda mylog.txt dosyasına “Wrong user” mesajı eklendi. Eğer doğru kullanıcı bilgileriyle giriş yapılırsa ise kullanıcının şehir araması yapabileceği SearchBar metoduna yönlendirildi. SearchBar metodu ise boş bir view’e (SearchBar.cshtml dosyasına) gitti. | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |
| --- | --- |
| **Index Metodu**      **Index.cshtml Dosyası**    **LoginUser Metodu** | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 29/07/2022** | | **KAPSAMI: Websitesinde SearchBar ve Şehir Listeleme İçin Gereken Interface ve Metotlar** |
|  | |  |
| * Login kısmında kullanıcı doğru bilgilerle giriş yaparsa websitesinde şehir araması yapabileceği bir arama kutucuğu gelir. Bunu SearchBar.cshtml dosyası sağladı. * Websitesinde kullanıcının otel araması için yazdığı şehir inputunu içeren modeli API’ye request olarak göndermek için bir Interface (ISearchingService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (SearchingService.cs) da oluşturuldu. * SearchBar.cshtml dosyası kullanıcıdan aldığı şehir inputunu HomeController içindeki GetArrivalAutoComplete metoduna gönderdi. GetArrivalAutoComplete metodu ise kullanıcı giriş yaptıktan sonra üretilen token ile şehir inputunu SearchingService.cs class’ındaki SearchCities metoduna gönderdi. SearchCities metodu şehir listesini almak için API’ye request attı. Request modelini oluşturmak için GetArrivalRequest isminde bir class oluşturuldu. Response modelini oluşturmak için GetArrivalResponse isminde bir class oluşturuldu. Response olarak aldığı şehirlerden Türkiye’de bulunanlar return edildi. Return edilen bu şehir listesini websitesinde bastırmak için GetArrivalAutoComplete metodu şehir listesini boş view’e (GetArrivalAutoComplete.cshtml dosyasına) gönderdi.   **SearchBar.cshtml**    **GetArrivalAutoComplete Metodu** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 02/08/2022** | | **KAPSAMI: Websitesinde Seçilen Şehirdeki Otelleri Listelemek İçin Gereken Interface ve Metotlar** |
|  | |  |
| * Websitesinde kullanıcının seçtiği şehrin id’sini içeren modeli API’ye request olarak göndermek için bir Interface (IPriceSearchingService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (IPriceSearchingService.cs) da oluşturuldu. * GetArrivalAutoComplete.cshtml dosyası kullanıcının seçtiği şehrin id’sini HomeController içindeki PriceSearch metoduna gönderdi. PriceSearch metodu ise kullanıcı giriş yaptıktan sonra üretilen token ile seçilen şehrin id’sini PriceSearchingService.cs class’ındaki PriceSearch metoduna gönderdi. PriceSearchingService.cs class’ının içerisindeki PriceSearch metodu otel listesini almak için API’ye request attı. Request modelini oluşturmak için PriceSearchRequest isminde bir class oluşturuldu. Response modelini oluşturmak için PriceSearchResponse isminde bir class oluşturuldu. Response olarak alınan otel listesini return edildi. Return edilen bu otel listesini websitesinde bastırmak için HomeController içindeki PriceSearch metodu otel listesini boş view’e (PriceSearch.cshtml dosyasına) gönderdi. Kullanıcı websitesindeki otellerden birine tıklayınca ise body’si boş GetProductInfo metodu çağırıldı.   **HomeController İçindeki PriceSearch Metodu**    **IPriceSearchingService.cs** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 03/08/2022** | | **KAPSAMI: Websitesinde Seçilen Şehirdeki Otelleri Listelemek İçin Gereken CsHtml Dosyası** |
|  | |  |
| * Kullanıcı otel rezervasyonu için arama yapmak istediği şehri seçtikten sonra websitesinde listelenen otellerden birine tıklayınca PriceSearch.cshtml dosyasından body’si boş GetProductInfo metodu çağırıldı.   **PriceSearch.cshtml**      **Websitesinde Şehir Listeleme Sayfası** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 04/08/2022** | | **KAPSAMI: Seçilen Şehirdeki Otel Listesinden Kullanıcının Herhangi Birini Seçip Detaylarını Görebilmesi İçin Gereken Interface ve Metotlar** |
|  | |  |
| * Websitesinde kullanıcının otel listesinden detaylı bilgilerini görmek istediği otelin id’sini ve offer id’sini içeren modeli API’ye request olarak göndermek için bir Interface (IProductInfoService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (ProductInfoService.cs) da oluşturuldu. * PriceSearch.cshtml dosyası kullanıcının detaylı bilgilerini görmek istediği otelin id’sini ve offer id’sini HomeController içindeki GetProductInfo metoduna gönderdi. GetProductInfo metodu ise kullanıcı giriş yaptıktan sonra üretilen token ile bilgileri görülmek istenen otelin id’sini ve offer id’sini ProductInfoService.cs class’ındaki GetProductInfo metoduna gönderdi. ProductInfoService.cs class’ı içerisindeki GetProductInfo metodu seçilen otelin detaylı bilgilerini almak için API’ye request attı. Request modelini oluşturmak için ProductInfoRequest isminde bir class oluşturuldu. Response modelini oluşturmak için ProductInfoResponse isminde bir class oluşturuldu. Response olarak alınan otelin detaylı bilgileri return edildi. Return edilen bu detaylı otel bilgisini websitesinde bastırmak için HomeController içindeki GetProductInfo metodu otel bilgisini boş view’e (GetProductInfo.cshtml dosyasına) gönderdi.   **HomeController İçindeki GetProductInfo Metodu**    **IProductInfoService.cs** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 05/08/2022** | | **KAPSAMI: Seçilen Şehirdeki Otel Listesinden Kullanıcının Herhangi Birini Seçip Detaylarını Görebilmesi İçin Gereken CsHtml Dosyası** |
|  | |  |
| * Kullanıcı otel rezervasyonu için arama yapmak istediği şehri seçtikten sonra websitesinde listelenen otellerden birine tıklayınca gelecek sayfayı oluşturmak için GetProductInfo.cshtml dosyası oluşturuldu.   **GetProductInfo.cshtml** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
|  | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 08/08/2022** | | **KAPSAMI: Websitede Rezervasyon Bilgilerinin Bastırılması İçin Gereken Interface ve Metotlar** |
|  | |  |
| * Websitesinde kullanıcının rezervasyon yapmak istediği otelin offer id’sini içeren modeli API’ye request olarak göndermek için bir Interface (IBeginTransactionService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (BeginTransactionService.cs) da oluşturuldu. Websitesinde rezervasyon yapan kullanıcının kişisel bilgilerini içeren modeli API’ye request olarak göndermek için bir Interface(ISetReservationService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class(SetReservationService.cs) da oluşturuldu. * Kullanıcı, detaylı bilgisini gördüğü otelden rezervasyon yaptırıp submit butonuna basınca kullanıcı bilgileri ve rezervasyon yapılan otelin offer id’si HomeController içindeki Booking metoduna gönderildi. Booking metodu transaction id’yi elde etmek için BeginTransactionService.cs class’ı içerisindeki BeginTransaction metoduna kullanıcı giriş yaptıktan sonra üretilen token ile offer id’yi gönderdi. BeginTransaction metodu transaction id’yi elde etmek için API’ye request attı. Request modelini oluşturmak için BeginTransactionRequest isminde bir class oluşturuldu. Booking metodu elde ettiği transaction id ve ayriyeten kullanıcı ismi, soyismi, mail’ini SetReservationService.cs class’ı içerisindeki SetReservation metoduna gönderdi. SetReservation metodu rezervasyon yapan kullanıcının rezervasyon bilgilerini almak için API’ye request attı. Request modelini oluşturmak için SetReservationRequest isminde bir class oluşturuldu. Response modelini oluşturmak için SetReservationResponse isminde bir class oluşturuldu. Response olarak alınan rezervasyon bilgileri return edildi. Return edilen bu rezervasyon bilgisini websitesinde bastırmak için HomeController içindeki Booking metodu rezervasyon bilgisini boş view’e (Booking.cshtml dosyasına) gönderdi.   **HomeController İçindeki Booking Metodu**    **IBeginTransactionService.cs** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
| **BeginTransactionService.cs İçindeki BeginTransaction Metodu**    **ISetReservationService.cs** | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 09/08/2022** | | **KAPSAMI: Websitede Rezervasyon Bilgilerinin Bastırılması İçin Gereken CsHtml Dosyası** |
|  | |  |
| * Kullanıcı seçtiği şehirdeki otel listesinden istediği oteli seçip rezervasyon yaptırdıktan sonra websitesinde rezervasyon bilgisinin olduğu sayfaya geçmek için Booking.cshtml dosyası oluşturuldu.   **Booking.cshtml**      **Kullanıcının Rezervasyon Yaparken Kendi Bilgilerini Girdiği Kısım** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 10/08/2022** | | **KAPSAMI: Rezervasyonun Tamamlanması İçin Gereken Interface ve Metotlar** |
|  | |  |
| * Websitesinde kullanıcının rezervasyon yapmak istediği otelin transaction id’sini içeren modeli API’ye request olarak göndermek için bir Interface (ICommitTransactionService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class (CommitTransactionService.cs) da oluşturuldu. Websitesinde rezervasyon yapan kullanıcının rezervasyon numarasını içeren modeli API’ye request olarak göndermek için bir Interface(IReservationDetailService.cs) oluşturuldu. Ayrıca, bu interface’i implement eden bir class(ReservationDetailService.cs) da oluşturuldu. * Rezervasyon bilgilerinin bastırıldığı sayfada kullanıcı Complete Reservation seçeneğine tıklarsa transaction id HomeController içindeki CompleteReservation metoduna gönderildi. CompleteReservation metodu rezervasyon numarasını elde etmek için CommitTransactionService.cs class’ı içerisindeki CompleteReservation metoduna kullanıcı giriş yaptıktan sonra üretilen token ile transaction id’yi gönderdi. CommitTransactionService.cs class’ı içerisindeki CompleteReservation metodu rezervasyon numarasanı elde etmek için API’ye request attı. Request modelini oluşturmak için CommitTransactionRequest isminde bir class oluşturuldu. HomeController içindeki CompleteReservation metodu elde ettiği rezervasyon numarası ve ayriyeten token’ı ReservationDetailService.cs class’ı içerisindeki SaveReservation metoduna gönderdi. SaveReservation metodu rezervasyon bilgilerini almak için API’ye request attı. Request modelini oluşturmak için ReservationDetailRequest isminde bir class oluşturuldu. Response modelini oluşturmak için ReservationDetailResponse isminde bir class oluşturuldu. Response olarak alınan rezervasyon bilgileri database’e kaydedilip return edildi. Return edilen bu rezervasyon bilgisini websitesinde bastırmak için HomeController içindeki CompleteReservation metodu rezervasyon bilgisini boş view’e (CompleteReservation.cshtml dosyasına) gönderdi.   **HomeController İçindeki CompleteReservation Metodu**    **ICommitTransactionService.cs** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
| **CommitTransaction.cs İçerisindeki CompleteReservation Metodu**    **IReservationDetailService.cs**    **ReservationDetailService.cs İçerisindeki SaveReservation Metodu** | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **YAPILAN İŞİN;** | | |
| **TARİHİ: 11/08/2022** | | **KAPSAMI: Rezervasyonun Tamamlanması İçin Gereken CsHtml Dosyası** |
|  | |  |
| **Rezervasyon Bilgilerinin Bastırıldığı Sayfa**     * Kullanıcı CompleteReservation yazısına tıklayınca bir sonraki gelecek sayfayı oluşturmak için CompleteReservation.cshtml dosyası yazıldı.   **CompleteReservation.cshtml** | | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** | |
|  |  | |

|  |  |
| --- | --- |
| * Kullanıcı CompleteReservation yazısına tıklarsa bir sonraki gelecek sayfa aşağıdaki gibidir. | |
| **Stajımı bu firmada yaptım.**  **Staj Yapanın İmzası** | **Staj Yeri Yetkilisinin**  **Adı, Soyadı, İmzası, Firma Kaşesi** |
|  |  |