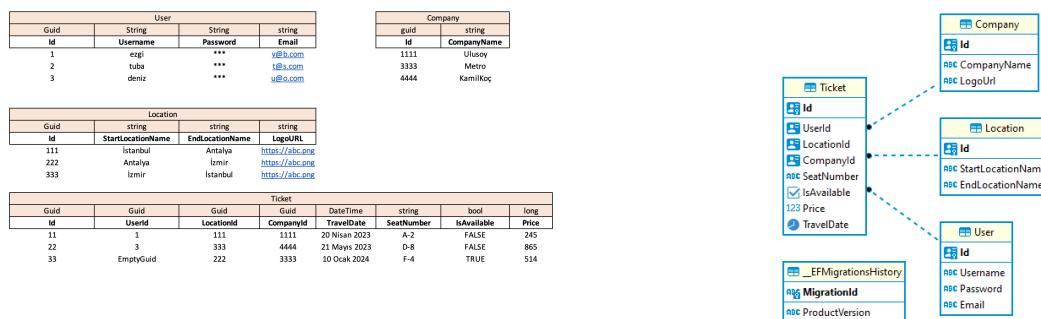


TicketBus

Ezgi Çalışkan/Tubanur Yazıcı

Veritabanı İlişkilendirmesi

Projemize aşağıda resmini paylaştığımız veritabanının tasarımını yaparak başladık. Temelde 4 adet tablo tasarımını yaptık ve bunları birbirleriyle ilişkilendirdik.



Bu tasarımından sonra .NET MVC Core 6 kullanarak projemizi geliştirmeye başladık. Projede codefirst yaklaşımını kullanarak aşağıdaki modelleri oluşturduk. Bu modelleri kullanarak veritabanı içerisinde migrationları yaptık.

The Models folder contains the following files:

- Account.cs
- Company.cs
- ErrorViewModel.cs
- Location.cs
- ManagementModel.cs
- SearchResults.cs
- Ticket.cs
- User.cs

Partial view: `~/Views/Shared/_Layout.cshtml`

```
1  using System.ComponentModel.DataAnnotations;
2  ...
3  namespace BusTicket_.Models
4  {
5      public class Ticket
6      {
7          public Guid Id { get; set; }
8          public Guid UserId { get; set; }
9          public Guid LocationId { get; set; }
10         public Guid CompanyId { get; set; }
11         public DateTime? TravelDate { get; set; }
12
13         [Required]
14         [StringLength(10)]
15         public string SeatNumber { get; set; }
16
17         [Required]
18         public long Price { get; set; }
19
20         [Required]
21         public bool IsAvailable { get; set; }
22
23         public virtual User User { get; set; }
24         public virtual Location Location { get; set; }
25         public virtual Company Company { get; set; }
26
27     }
28
29 }
30
31 }
```

Kod içerisindeki ilişkilendirmeyi context klasörü altındaki classlar içerisinde yaptık.

```
16    public static void MigrateForMssql()
17    {
18        var app = ConfigurationBuilder.CreateInMemoryBuilder();
19
20        var connectionStringUtility = new ConnectionStringUtility(app);
21        var connectionStrings = connectionStringUtility.GetConnectionString("Default");
22        var defaultConnectionString = connectionStrings["Default"];
23        var defaultConnection = defaultConnectionString.Create();
24
25        if (defaultConnection != null)
26        {
27            using (var serviceScope = defaultConnection.GetServiceScope())
28            {
29                var db = serviceScope.ServiceProvider.GetService<DbContext>();
30                db.Database.Migrate();
31            }
32        }
33    }
34
35    default:
36    {
37        throw new Exception("Connection for unknown DB: Check connection string.");
38    }
39
40 }
```

```

1  using BusTicket_.Context;
2  using Microsoft.EntityFrameworkCore;
3  using Microsoft.EntityFrameworkCore.Infrastructure;
4  using Microsoft.EntityFrameworkCore.Metadata;
5  using Microsoft.EntityFrameworkCore.Migrations;
6  using Microsoft.EntityFrameworkCore.Storage;
7  using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
8  using System;
9  using System.Collections.Generic;
10 using System.Linq;
11 using System.Reflection;
12 using System.Threading.Tasks;
13
14 namespace BusTicket_.Context
15 {
16     public class ApplicationDbContext : DbContext
17     {
18         public DbSet<User> User { get; set; }
19         public DbSet<Company> Company { get; set; }
20         public DbSet<Ticket> Ticket { get; set; }
21         protected override void OnModelCreating(ModelBuilder modelBuilder)
22         {
23             modelBuilder.ApplyConfiguration(new UserEntityTypeConfiguration());
24             modelBuilder.ApplyConfiguration(new CompanyEntityTypeConfiguration());
25             modelBuilder.ApplyConfiguration(new TicketEntityTypeConfiguration());
26         }
27     }
28 }

```

Migrationları yaparken aşağıdaki ServiceCollectionsExtensions.cs içerisindeki MigrateForMultiDb methodunu kullandık.

```

1  using BusTicket_.Constants;
2  using BusTicket_.Utilities;
3  using Microsoft.AspNetCore.Builder;
4  using Microsoft.EntityFrameworkCore;
5  using Microsoft.Extensions.Configuration;
6  using Microsoft.Extensions.DependencyInjection;
7  using System;
8  using System.Collections.Generic;
9  using System.Linq;
10 using System.Reflection;
11 using System.Threading.Tasks;
12
13 namespace BusTicket_.Extensions
14 {
15     public static class ServiceCollectionExtensions
16     {
17         public static void AddDbContextForMultiDb<T>(
18             this IServiceCollection services, IConfiguration configuration, string assemblyName)
19             where T : DbContext
20         {
21
22             var connectionStringUtility = new ConnectionStringUtility(configuration);
23             var connectionString = connectionStringUtility.GetConnectionStringFromConfig();
24             var defaultConnectionName = connectionStringUtility.GetDefaultConnectionStringName();
25             services.AddDbContext<T>(options =>
26             {
27                 switch (defaultConnectionName)
28                 {
29                     case ConnectionStringConstants.PostgreSqlConnection:
30                         options.UseNpgsql(
31                             connectionString,
32                             o =>
33                             {
34                                 o.MigrationsAssembly(assemblyName);
35                                 o.EnableRetryOnFailure(maxRetryCount: 10, maxRetryDelay: TimeSpan.FromSeconds(30),
36                                 errorCodesToAdd: null);
37                             });
38                         break;
39
40                     default:
41                         throw new Exception("Connection For Unknown Db! Check connection string.");
42                 }
43             });
44         }
45     }

```

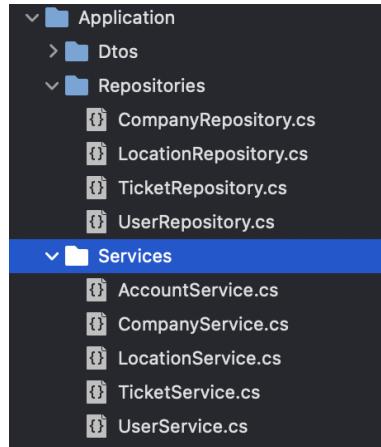
Ardından Controllerleri yazarak devam ettik. Temelde kullandığımız HomeController.cs içerisindenden aşağıda bazı actionları görebilirsiniz. Bu actionlar ile sayfalar arasında geçişleri ve gerekli yerlerde servis ve repositoryleri kullanarak veritabanında Veri alma/yazma işlemlerini gerçekleştirdik.

```

1  using BusTicket_.Application;
2  using BusTicket_.Application.Services;
3  using BusTicket_.Domain.Services;
4  using Microsoft.AspNetCore.Mvc;
5  using Microsoft.AspNetCore.Mvc.RazorPages;
6  using Microsoft.AspNetCore.Mvc.Rendering;
7  using Microsoft.Extensions.Logging;
8  using System.Diagnostics;
9
10 namespace BusTicket_.Controllers
11 {
12     [Route("home")]
13     public class HomeController : Controller
14     {
15         private readonly ILogger<HomeController> _logger;
16         private readonly CompanyService _companyService;
17         private readonly TicketService _ticketService;
18
19         public HomeController(
20             ILogger<HomeController> logger,
21             CompanyService companyService,
22             TicketService ticketService)
23         {
24             _logger = logger;
25             _companyService = companyService;
26             _ticketService = ticketService;
27         }
28
29         // GET: Home/Index
30         [HttpGet("~/~")]
31         [HttpGet("~/Index")]
32         [HttpGet("~/Index")]
33         public async Task<ActionResult> Index()
34         {
35             if (true)
36             {
37                 return RedirectToAction("Ticket", "Home");
38             }
39         }
40     }
41 }

```

Herbir veritabanı tablomuz için o tabloyla bağlantı kuran servis ve repository'leri yazdık. Bunlar sayesinde gerekli tablolardan verileri çekme ve yazma işlemlerini yaptık. Toplamda 4 servis ve 4 repository classı interfaceleri ile birlikte oluşturduk. Bunlardan bazılarının örneklerini aşağıda bulabilirsiniz.



```

1  using Microsoft.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore.ChangeTracking;
3  using Microsoft.EntityFrameworkCore.DomainServices;
4  using Microsoft.EntityFrameworkCore.EntityProxies;
5  using Microsoft.EntityFrameworkCore.InMemory;
6  using System;
7  using System.Linq;
8  using System.Threading;
9  using System.Threading.Tasks;
10 using System.Linq.Expressions;
11 using System.Collections.Generic;
12 using System.Linq;
13
14 namespace TicketBus.Application.Repositories
15 {
16     public class TicketRepository : ITicketRepository
17     {
18         private readonly ApplicationDbContext _cts;
19         public TicketRepository(ApplicationDbContext cts)
20         {
21             _cts = cts;
22         }
23
24         public async Task<Ticket> AddAsync(Ticket entity)
25         {
26             var result = _cts.Ticket.Add(entity);
27             await _cts.SaveChangesAsync();
28             return result.Entity.Id;
29         }
30
31         public async Task<Ticket> DeleteAsync(Ticket entity)
32         {
33             var result = _cts.Ticket.Remove(entity);
34             await _cts.SaveChangesAsync();
35             return result.Entity.Id;
36         }
37
38         public async Task<Ticket> DetachAsync(Ticket entity)
39         {
40             var result = _cts.Ticket.Attach(entity);
41             await _cts.SaveChangesAsync();
42             return result.Entity.Id;
43         }
44
45         public async Task<Ticket> EditAsync(Ticket entity)
46         {
47             var result = _cts.Ticket.Update(entity);
48             await _cts.SaveChangesAsync();
49             return result.Entity.Id;
50         }
51
52         public async Task<Ticket> GetAsync(Guid id)
53         {
54             var query = _cts.Ticket.AsNoTracking();
55
56             if (id != null) && id != Guid.Empty)
57             {
58                 query = query.Where(x => x.Id == id);
59             }
60             if (_cts.LocationId != Guid.Empty)
61             {
62                 query = query.Where(x => x.LocationId == _cts.LocationId);
63             }
64             if (_cts.ClientId != Guid.Empty)
65             {
66                 query = query.Where(x => x.ClientId == _cts.ClientId);
67             }
68             if (_cts.TicketNumber != null)
69             {
70                 query = query.Where(x => x.TicketNumber == _cts.TicketNumber);
71             }
72             if (_cts.Expiration != null)
73             {
74                 query = query.Where(x => x.Expiration == _cts.Expiration);
75             }
76             if (_cts.InvoiceNumber != null)
77             {
78                 query = query.Where(x => x.InvoiceNumber == _cts.InvoiceNumber);
79             }
80             if (_cts.InvoiceStatus != null)
81             {
82                 query = query.Where(x => x.InvoiceStatus == _cts.InvoiceStatus);
83             }
84             if (_cts.InvoiceValue != null)
85             {
86                 query = query.Where(x => x.InvoiceValue == _cts.InvoiceValue);
87             }
88             if (_cts.StockValue != null)
89             {
90                 query = query.Where(x => x.StockValue == _cts.StockValue);
91             }
92             if (_cts.StockNumber != null)
93             {
94                 query = query.Where(x => x.StockNumber == _cts.StockNumber);
95             }
96             if (_cts.TicketType != null)
97             {
98                 query = query.Where(x => x.TicketType == _cts.TicketType);
99             }
100            if (_cts.TicketStatus != null)
101            {
102                query = query.Where(x => x.TicketStatus == _cts.TicketStatus);
103            }
104        }
105    }
106}

```

Anasayfa tasarımını ve kullanıcı görüntüsünü aşağıda paylaşıyorum.

BusTicket Login

Bizimle Seyahate Hazır mısınız?

Kalkış Yeri: Nereden... Varış Yeri: Nereye...
Firma İsmi: Firma İsmi... Seyahat Tarihi: gg.aaaaaa

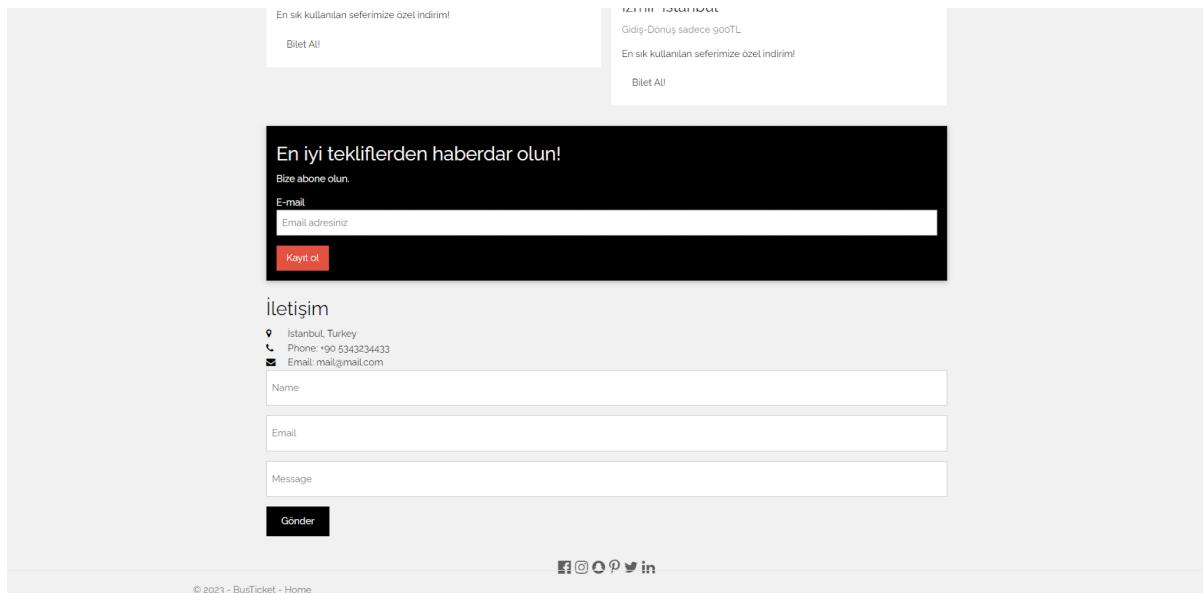
Varan	5/1/2023 4:00:00 PM Kalkış Yeri: Bursa <input type="button" value="A1"/> <input type="button" value="Satın Al"/>	Varış Yeri: Kars	563 ⁰⁰ ₺
Kamil Koç	5/3/2023 12:45:00 PM Kalkış Yeri: Balıkesir <input type="button" value="A1"/> <input type="button" value="Satın Al"/>	Varış Yeri: Bodrum	425 ⁰⁰ ₺

Kamil Koç	Balıkesir <input type="button" value="A1"/> <input type="button" value="Satın Al"/>	Varış Yeri: Bodrum	515 ⁰⁰ ₺
METRO	4/29/2023 11:50:00 PM Kalkış Yeri: İstanbul <input type="button" value="A1"/> <input type="button" value="Satın Al"/>	Varış Yeri: Antalya	515 ⁰⁰ ₺

Popüler Rotalar

Kaş/Antalya	Izmir	Ankara
Mardin	Bolu	

Popüler Seferler



```
    private void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final Button button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

```

    item arguments="parent" method="post">
        <input type="text" value="1" style="width:100px; height:30px; border:none; border-radius:5px; font-size:14px; margin-bottom:10px;" />
        <input type="button" value="提交" style="width:100px; height:30px; border:none; border-radius:5px; background-color:#007bff; color:white; font-weight:bold; font-size:14px; margin-top:10px;" />
    
```

```

        <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: white; color: black; z-index: 1000000; margin: auto; border-radius: 10px; padding: 10px; border: 1px solid #ccc; font-size: 14px; font-family: sans-serif; box-sizing: border-box; border: none; background-clip: padding-box; transition: all 0.3s ease-in-out; opacity: 0; visibility: hidden; transform: rotate(-15deg);>
            <div style="margin-bottom: 10px; border-bottom: 1px solid #ccc; padding-bottom: 5px; font-weight: bold; font-size: 1.2em; text-align: center; position: relative; width: 100%;">
                <div style="display: flex; justify-content: space-between; align-items: center; width: 100%; position: relative; height: 100%;>
                    <div style="flex-grow: 1; text-align: center; font-size: 0.9em; position: relative; width: 100%; height: 100%;>
                        <div style="display: flex; justify-content: space-around; align-items: center; width: 100%; height: 100%;>
                            <div style="text-align: center; width: 100px; height: 100px; position: relative; border: 1px solid #ccc; border-radius: 50%; background-color: #f0f0f0; display: flex; align-items: center; justify-content: center; font-size: 1.2em; font-weight: bold; font-family: sans-serif; font-style: italic; width: 100%; height: 100%;>
                                <span>OK</span>
                            </div>
                            <div style="width: 100px; height: 100px; position: relative; border: 1px solid #ccc; border-radius: 50%; background-color: #f0f0f0; display: flex; align-items: center; justify-content: center; font-size: 1.2em; font-weight: bold; font-family: sans-serif; font-style: italic; width: 100%; height: 100%;>
                                <span>Cancel</span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```
<div class="row justify-content-center" style="margin-top: 10px">
    <div class="col-12 col-md-8 col-lg-6" style="text-align: center">
        <img alt="Screenshot of the 'New User' registration page showing the 'Forgot Password?' link." data-bbox="380 117 610 210" href="https://www.simplenoteapp.com/auth/signup?next=/users/forgot-password"/>
        <p>Forgot Password?</p>
    </div>
</div>

<div class="row justify-content-center" style="margin-top: 10px">
    <div class="col-12 col-md-8 col-lg-6" style="text-align: center">
        <img alt="Screenshot of the 'Forgot Password' page showing the 'Reset Password' button." data-bbox="380 210 610 300" href="https://www.simplenoteapp.com/auth/forgot-password?token=69e75a0250f810ff9709d3c9"/>
        <p>Reset Password</p>
    </div>
</div>
```

```
177 <div class="content">
178   <div class="card">
179     <div class="card-content">
180       <div class="card-media">
181         
182       <div class="card-text">
183         <div class="text-item">Title: The Great Gatsby</div>
184         <div class="text-item">Author: F. Scott Fitzgerald</div>
185         <div class="text-item">Genre: Novel (Fiction)</div>
186         <div class="text-item">Rating: 4.5/5 (Based on 100,000 reviews)</div>
187         <div class="text-item">Buy Now</div>
188       </div>
189     </div>
190   </div>
191 </div>
192 <div class="content">
193   <div class="card">
194     <div class="card-content">
195       <div class="card-media">
196         
197       <div class="card-text">
198         <div class="text-item">Title: The Great Gatsby</div>
199         <div class="text-item">Author: F. Scott Fitzgerald</div>
200         <div class="text-item">Genre: Novel (Fiction)</div>
201         <div class="text-item">Rating: 4.5/5 (Based on 100,000 reviews)</div>
202         <div class="text-item">Buy Now</div>
203       </div>
204     </div>
205   </div>
206 </div>
```

Bilet Filtreleme;

The screenshot shows a travel booking interface for a bus service. At the top, there's a search bar with the text "Bizimle Seyahate Hazır mısınız?". Below it, the departure location is set to "İstanbul" and the arrival location is "Nereye..". The travel date is "99.aaaaaa". A dropdown menu for seat selection shows options like A1, A2, A3, A4, B5, B6, B7, B8, C9, C10, C11, C12, D13, D14, D15, D16, E17, E18, E19, E20, and C11. The search button is labeled "Bilet Ara".

Below the search bar, two ticket options are displayed:

- Varan**: Departure time is 5/1/2023 4:00:00 PM. The ticket price is 563 TL.
- METRO**: Departure time is 4/29/2023 11:50:00 PM. The ticket price is 515 TL.

At the bottom, there's a section titled "Popüler Rotalar" (Popular Routes) featuring three thumbnail images of scenic locations.

This screenshot shows a similar travel booking interface. The search bar at the top asks "Bizimle Seyahate Hazır mısınız?". The departure location is "İstanbul" and the arrival location is "Nereye..". The travel date is "gg.aaaaaa". A dropdown menu for seat selection shows options like A1, A2, A3, A4, B5, B6, B7, B8, C9, C10, C11, C12, D13, D14, D15, D16, E17, E18, E19, E20, and C11. The search button is labeled "Bilet Ara".

Below the search bar, one ticket option is displayed:

- METRO**: Departure time is 4/29/2023 11:50:00 PM. The ticket price is 515 TL.

At the bottom, there's a section titled "Popüler Rotalar" (Popular Routes) featuring three thumbnail images of scenic locations.

The screenshot shows the BusTicket login page with a search interface overlaid on a blurred background image of a city street at night. The search fields include 'Kalkış Yeri' (Istanbul), 'Varış Yeri' (Nereye...), 'Firma İsmi' (Firma İsmi...), and 'Seyahat Tarihi' (03.05.2023). A 'Bilet Ara' (Search Ticket) button is present. Below the search bar, two route options are listed:

- KamilKoç**: Kalkış Yeri (Balıkesir) - Varış Yeri (Bodrum) on 5/3/2023 at 12:45:00 PM. Price: 425,00 ₺.
- varan**: Kalkış Yeri (Bursa) - Varış Yeri (Kars) on 5/3/2023 at 4:00:00 PM. Price: 563,00 ₺.

Kullanıcı girişi sayfası tasarımı ve kullanıcı görüntüsü ;

The screenshot shows the 'Kullanıcı Giriş'i (User Login) form. It includes fields for 'Kullanıcı Adı' (Username) and 'Şifre' (Password), both with placeholder text. A 'Giriş Yap' (Login) button is below the password field. At the bottom, there are links for 'Beni Hatırla' (Remember Me) and 'İptal' (Cancel), along with a link to 'Şifremi mi unuttum?' (Forgot my password?).

© 2023 - BusTicket - [Home](#)

```

1  ViewData["Title"] = "Login";
2
3
4
5  <head>
6    <meta name="viewport" content="width=device-width, initial-scale=1">
7    <title> Gözüp Yapan </title>
8    <style>
9
10   button {
11     background-color: lightgray;
12     width: 100px;
13     height: 30px;
14     padding: 10px;
15     margin: 10px 0px;
16     border: 2px solid green;
17     border-radius: 10px;
18     cursor: pointer;
19   }
20
21   form {
22     border: 2px solid #f1f1f1;
23   }
24
25   input[type="text"], input[type="password"] {
26     width: 100px;
27     margin: 10px 0px;
28     padding: 10px 20px;
29     display: inline-block;
30     border: 2px solid green;
31     border-radius: 10px;
32   }
33
34   button:hover {
35     opacity: 0.7;
36   }
37
38   .cancelbtn {
39     width: auto;
40     padding: 10px 20px;
41     margin: 10px 0px;
42   }

```

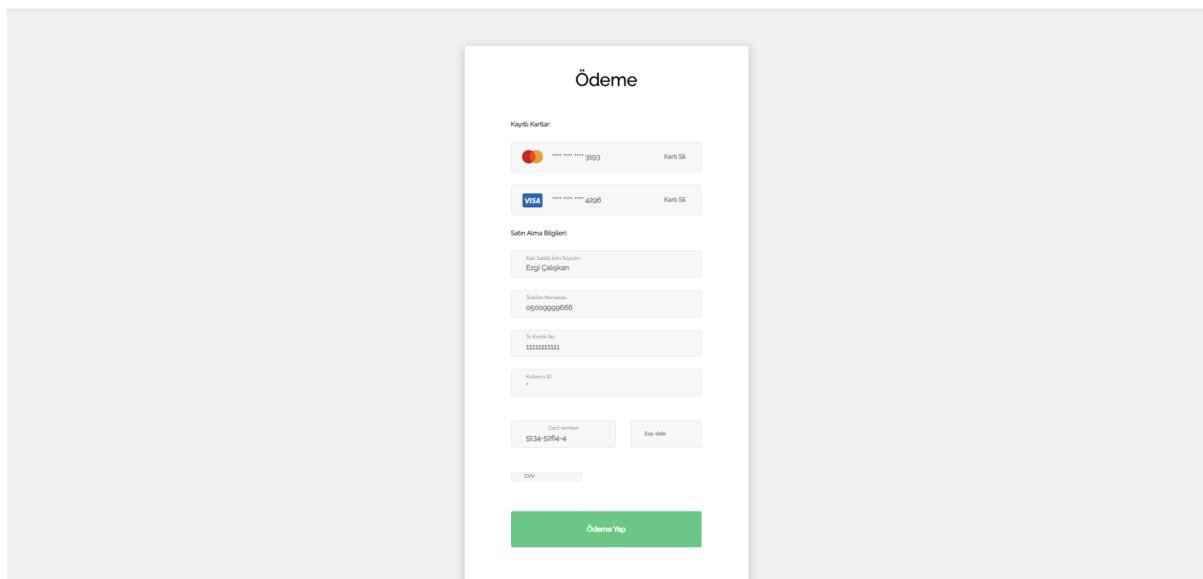
```

49   </style>
50   </head>
51   <body>
52     <center> <h2> Kullanıcı Giriş'i </h2> </center>
53     <form>
54       <div class="contain">
55         <label>Kullanıcı Adı : </label>
56         <input type="text" placeholder="" name="username" required>
57         <label>Şifre : </label>
58         <input type="password" placeholder="" name="password" required>
59         <input type="checkbox" checked="checked"> Beni Hatırla
60         <button type="button" class="cancelbtn" Iptal></button>
61         <a href="#">Şifremi mi unuttum? </a>
62       </div>
63     </form>
64   </body>
65
66
67

```

Satın Alma sayfası;

BusTicket Login



```
119 // .card {
120 //   margin: auto;
121 //   width: 60px;
122 //   padding: 1.5rem;
123 //   box-shadow: 0 6px 20px 0 rgba(0, 0, 0, 0.19);
124 //
125 //   .at-50 {
126 //     margin-top: 75px;
127 //   }
128 //
129 //   .at-50 {
130 //     margin-bottom: 56px;
131 //   }
132 //
133 //   @media (max-width:767px) {
134 //     .card {
135 //       width: 90%;
136 //       padding: 1.5rem;
137 //     }
138 //
139 //     @media (width:366px) {
140 //       .card {
141 //         width: 50%;
142 //         padding: 8vh;
143 //       }
144 //
145 //       .card-title {
146 //         font-weight: 700;
147 //         font-size: 2.5em;
148 //       }
149 //
150 //       .nav {
151 //         display: flex;
152 //       }
153 //     }
154 //   }
155 //
156 //   .nav {
157 //     display: flex;
158 //   }
159 // }
```

```
148 = .nav li {  
149     list-style-type: none;  
150     display: inline-block;  
151     vertical-align: unset;  
152     margin-bottom: 6vh;  
153 }  
154 = .nav li i {  
155     padding: 1rem;  
156 }  
157 = .nav li a {  
158     color: black;  
159     text-decoration: none;  
160 }  
161 = .active {  
162     border-bottom: 2px solid black;  
163     font-weight: bold;  
164 }  
165 = input {  
166     border: none;  
167     outline: none;  
168     font-size: 1rem;  
169     font-weight: 600;  
170     color: grey;  
171     width: 100%;  
172     min-width: 100%;  
173     background-color: transparent;  
174     border-color: transparent;  
175     margin: 0;  
176 }  
177 = form {  
178     color: grey;  
179     border: none;  
180     font-size: 0.875em;  
181     font-weight: bold;  
182 }
```

```
        label {
            display: flex;
            flex-direction: column;
            align-items: center;
            text-align: center;
            font-size: 14px;
            font-weight: bold;
            color: #333;
            margin-bottom: 10px;
        }

        input {
            width: 100px;
            height: 30px;
            border-radius: 10px;
            border: none;
            outline: none;
            background-color: #f0f0f0;
            padding: 5px;
            font-size: 14px;
            font-family: Arial, sans-serif;
        }

        button {
            width: 100px;
            height: 30px;
            border-radius: 10px;
            border: none;
            outline: none;
            background-color: #007bff;
            color: white;
            font-size: 14px;
            font-family: Arial, sans-serif;
            font-weight: bold;
            text-decoration: none;
            cursor: pointer;
        }
    
```

Biletlerim sayfası;

Biletlerim

 KAMILKOÇ KamilKoç	4/26/2023 1:00:00 PM	Kalkış Yeri Kaş	İptal Et	Verüş Yeri Ardahan	465 ⁰⁰
 SAIMEDY SAIMEDY	4/28/2023 8:30:00 PM	Kalkış Yeri İstanbul	İptal Et	Verüş Yeri Mardin	345 ⁰⁰

© 2023 - BusTicket - [Home](#)