



ITAN

КОИ СМЕ НИЕ

Ние сме най – модерният шоурум за отдаване на коли под наем за луксозни автомобили в България.

Представяме качество, надеждност, лукс. Имаме представителства в шестте най – големи български града, като главния е в София.



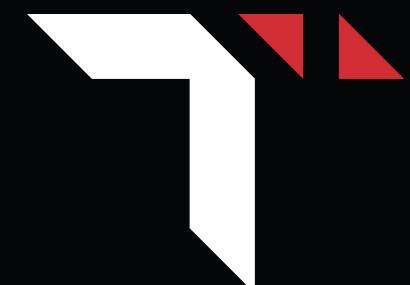
ЦЕЛ НА СОФТУЕР

Създадохме софутер, който може да се използва ще се използва от нашата компания ТИТАН. Софтуера е админски, като целта му е да подпомогне продажбения процес.



ИЗГОТВИЛИ

Разпределение на ролите:
Пламен Желев- дизайн на сайта, създаване на кода за таблиците,
създаване на презентацията.
Николай Станев- свързване на таблиците в базата данни в релация,
писане на компонентни тестове, дизайн на сайта.
Останалата част е обща работа.



РЕАЛИЗАЦИЯ

1. Нахвърляне идеи на приложението
2. Разучаване рентиерската индустрия
3. Създаване на моделите на БД
4. Генерираме таблици чрез EF
5. ASP.Net Core Blazor Pages
6. ASP.Net cold generator-генерира страницата на сайта
7. Подобрене на автоматично генерираните страници
8. Добаваш примерни данни
- 9.Проверки и допълнителни функции



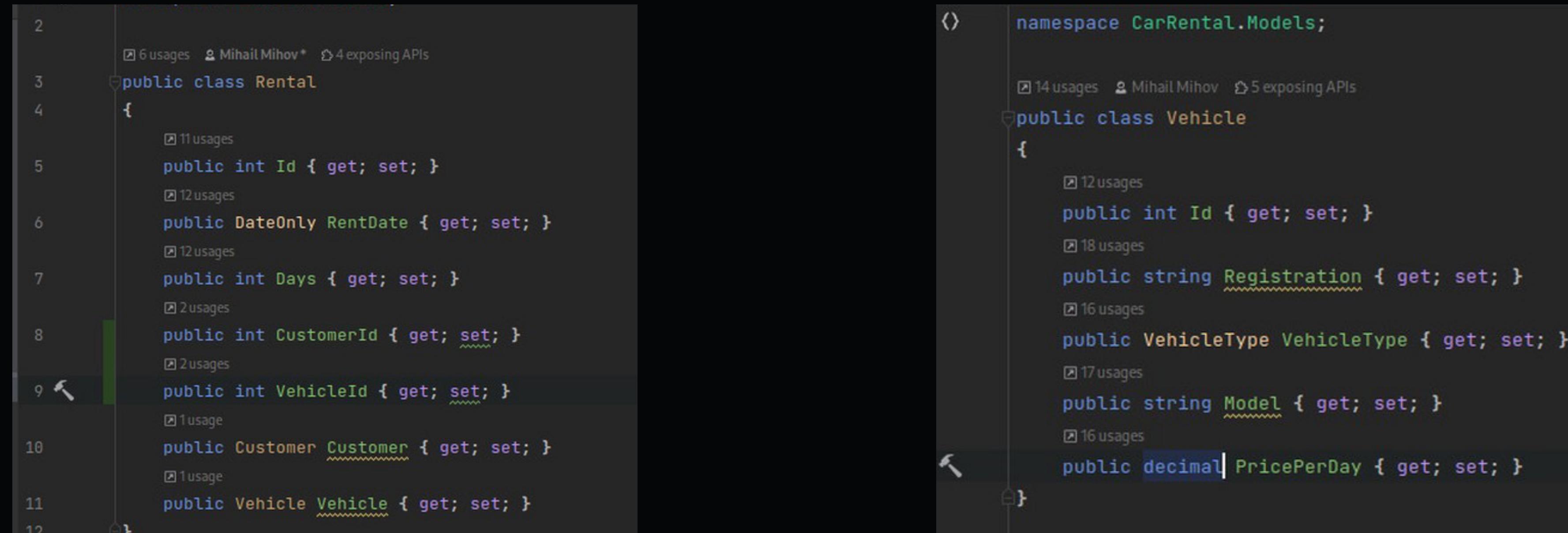
ТЕХНОЛОГИИ

Използвани технологии за проекта:

Entity Framework
ASP.NET Core
Blazor Pages
SQLite
Bootstrap
JQuery



РЕАЛИЗАЦИЯ НА ПРИЛОЖЕНИЕТО



The image shows two side-by-side code editors. The left editor displays the `Rental` class, and the right editor displays the `Vehicle` class. Both classes are part of the `CarRental.Models` namespace and include properties for `Id`, `RentDate`, `Days`, `CustomerId`, `VehicleId`, `Customer`, and `Vehicle`. The `Vehicle` class also includes properties for `Registration`, `VehicleType`, `Model`, and `PricePerDay`.

```
2     ↗ 6 usages  ↗ Mihail Mihov*  ↗ 4 exposing APIs
3     ↗ public class Rental
4     {
5         ↗ 11 usages
6             ↗ public int Id { get; set; }
7             ↗ 12 usages
8                 ↗ public DateOnly RentDate { get; set; }
9                 ↗ 12 usages
10                ↗ public int Days { get; set; }
11                ↗ 2 usages
12                    ↗ public int CustomerId { get; set; }
13                    ↗ 2 usages
14                        ↗ public int VehicleId { get; set; }
15                        ↗ 1 usage
16                            ↗ public Customer Customer { get; set; }
17                            ↗ 1 usage
18                                ↗ public Vehicle Vehicle { get; set; }
```

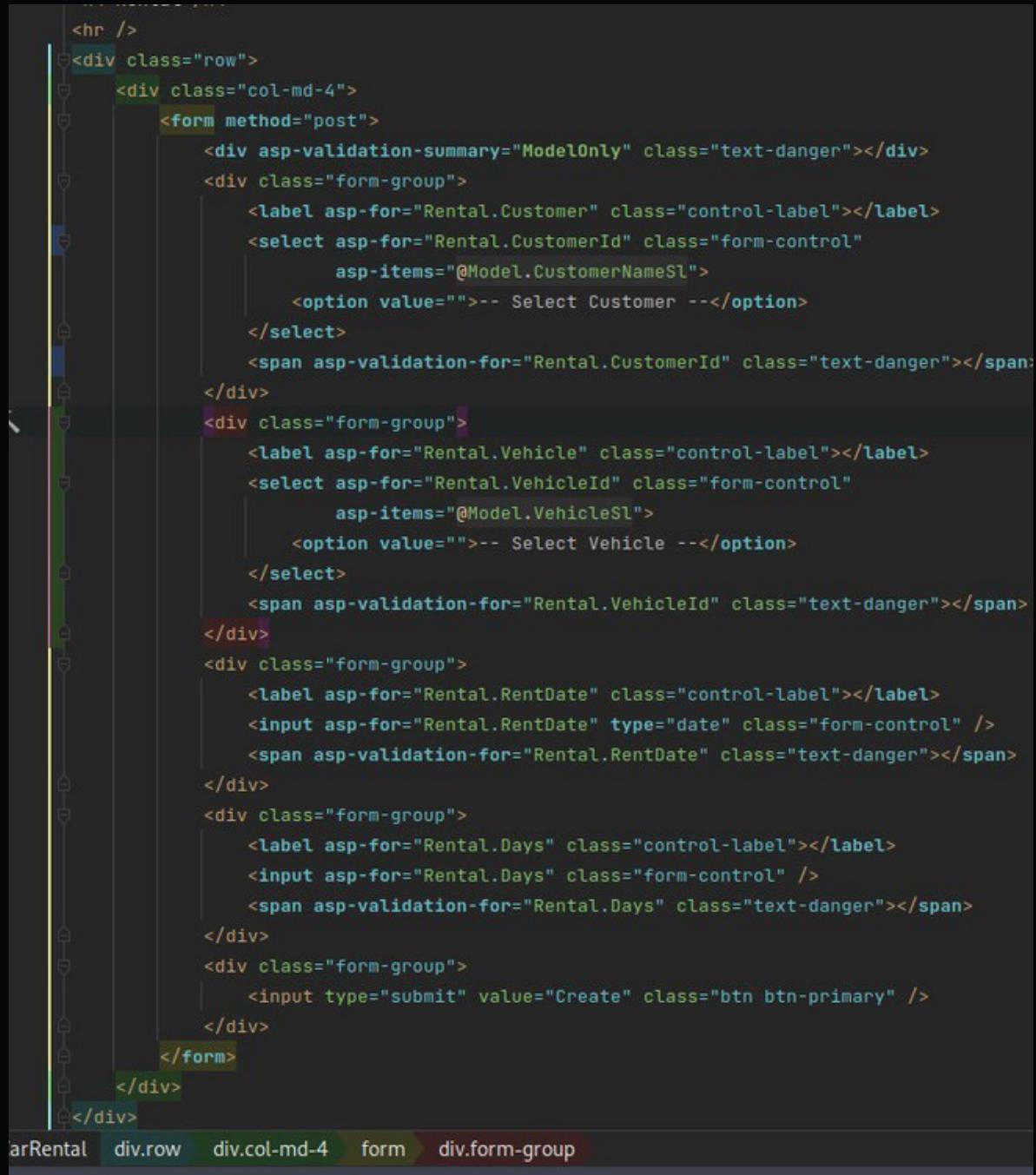
```
.namespace CarRental.Models;
14 usages  ↗ Mihail Mihov  ↗ 5 exposing APIs
public class Vehicle
{
    ↗ 12 usages
        ↗ public int Id { get; set; }
        ↗ 18 usages
            ↗ public string Registration { get; set; }
            ↗ 16 usages
                ↗ public VehicleType VehicleType { get; set; }
                ↗ 17 usages
                    ↗ public string Model { get; set; }
                    ↗ 16 usages
                        ↗ public decimal PricePerDay { get; set; }
```

Таблиците се генерираят автоматично от тези модели на Entity Framework

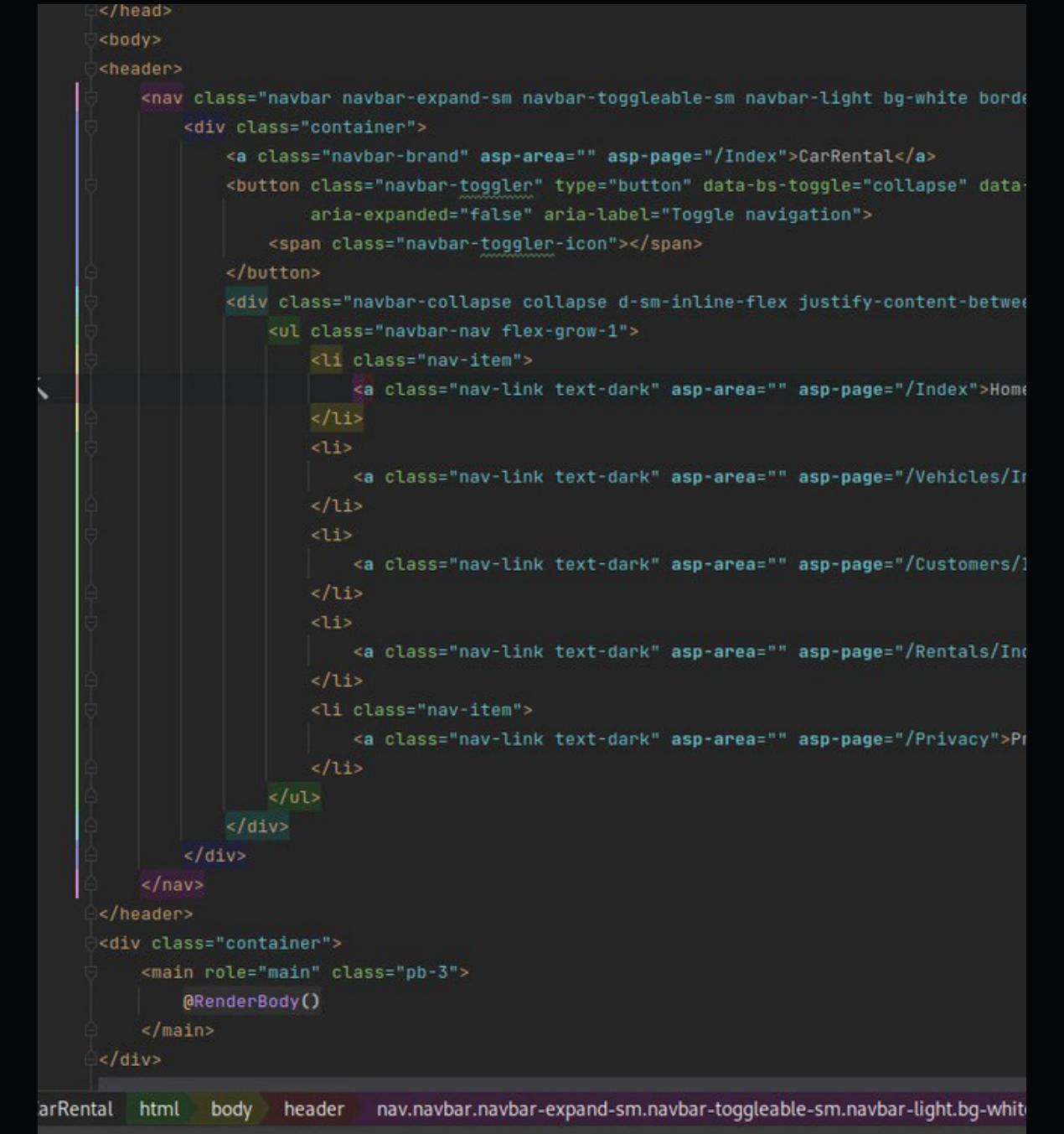


РЕАЛИЗАЦИЯ НА ПРИЛОЖЕНИЕТО

Страница за създаване
на нов рентал.



```
<hr />
<div class="row">
  <div class="col-md-4">
    <form method="post">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="Rental.Customer" class="control-label"></label>
        <select asp-for="Rental.CustomerId" class="form-control" asp-items="@Model.CustomerNameSl">
          <option value="">-- Select Customer --</option>
        </select>
        <span asp-validation-for="Rental.CustomerId" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Rental.Vehicle" class="control-label"></label>
        <select asp-for="Rental.VehicleId" class="form-control" asp-items="@Model.VehicleSl">
          <option value="">-- Select Vehicle --</option>
        </select>
        <span asp-validation-for="Rental.VehicleId" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Rental.RentDate" class="control-label"></label>
        <input asp-for="Rental.RentDate" type="date" class="form-control" />
        <span asp-validation-for="Rental.RentDate" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Rental.Days" class="control-label"></label>
        <input asp-for="Rental.Days" class="form-control" />
        <span asp-validation-for="Rental.Days" class="text-danger"></span>
      </div>
      <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```



```
</head>
<body>
<header>
  <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow pb-3" asp-area="" asp-page="/Index">
    <div class="container">
      <a class="navbar-brand" asp-area="" asp-page="/Index">CarRental</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between" id="navbarSupportedContent">
        <ul class="navbar-nav flex-grow-1">
          <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
          </li>
          <li>
            <a class="nav-link text-dark" asp-area="" asp-page="/Vehicles/Index">Vehicles</a>
          </li>
          <li>
            <a class="nav-link text-dark" asp-area="" asp-page="/Customers/Index">Customers</a>
          </li>
          <li>
            <a class="nav-link text-dark" asp-area="" asp-page="/Rentals/Index">Rentals</a>
          </li>
          <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-page="/Privacy">Privacy</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>
<div class="container">
  <main role="main" class="pb-3">
    @RenderBody()
  </main>
</div>
```

Началната страница
и под менюта



РЕАЛИЗАЦИЯ НА ПРИЛОЖЕНИЕТО

```
3  var builder = WebApplication.CreateBuilder(args);
4
5  builder.Services.AddRazorPages();
6  builder.Services.AddDbContext<RentalContext>(<optionsAction: options =>
7      options.UseSqlite(builder.Configuration.GetConnectionString(name: "RentalContext")) ?? throw new InvalidOperationException
8
9  var app:WebApplication = builder.Build();
10
11 using (var scope = app.Services.CreateScope())
12 {
13     var services:ServiceProvider = scope.ServiceProvider;
14     DbInitializer.Initialize(services);
15 }
16
17 if (!app.Environment.IsDevelopment())
18 {
19     app.UseExceptionHandler("/Error");
20     app.UseHsts();
21 }
22
23 app.UseHttpsRedirection();
24 app.UseStaticFiles();
25
26 app.UseRouting();
27
28 app.UseAuthorization();
29
30 app.MapRazorPages();
31
32 app.Run();
```

Инициализация и връзка с базата данни



**БЛАГОДАРЯ ЗА
ВНИМАНИЕТО**

