**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**Дисциплина:** BackEnd-разработка

**Тема:** Добавление в веб-приложение на основе ASP.NET Core маршрутизации.

**Выполнил: студент группы 231-332**

**Чесноков Александр Евгеньевич**

(Фамилия И.О. обучающегося)

**Дата, подпись** 01.10.2025 

(Дата)(Подпись)

**Проверил:** \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_*

(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_ \_\_\_\_\_\_\_\_*

(Дата)(Подпись)

**Москва**

**2025**

**Цель:**

Познакомиться с основами настройки и использования маршрутизации в веб-приложениях на основе ASP.NET Core.

**Ход работы:**

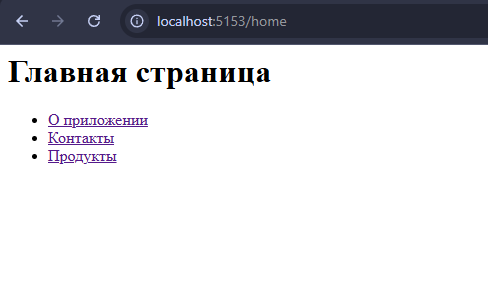
Первым шагом было создание проекта ASP.NET Core Web App в Visual Studio. Далее я определил контроллеры и маршруты, обеспечив удобную навигацию между страницами через HTML-ссылки.

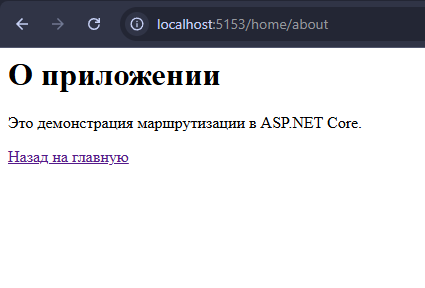
Я реализовал два контроллера: HomeController и ProductController. HomeController содержит методы Index() и About(). Метод Index() отображает главную страницу с приветственным сообщением и ссылками на остальные разделы приложения. Метод About() возвращает информацию о приложении. Для обеих страниц я использовал HTML-контент, что позволяет пользователю перемещаться по страницам с помощью кликабельных ссылок, делая навигацию более удобной и наглядной. Для маршрутизации применялся атрибут [Route], что позволило явно задавать URL-адреса методов контроллера и обеспечивать удобный доступ к страницам.

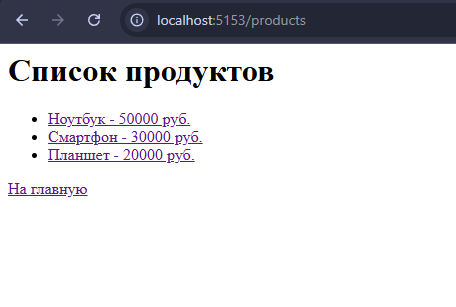
ProductController демонстрирует работу с параметрами маршрутов. В нём реализованы методы Index(), Details(int id) и Filter(int? minPrice). Метод Index() выводит список всех продуктов с гиперссылками на их страницы деталей. Метод Details(int id) отображает информацию о выбранном продукте по его идентификатору и включает проверку на существование продукта, возвращая страницу с сообщением об ошибке, если продукта нет. Метод Filter(int? minPrice) позволяет фильтровать продукты по минимальной цене, демонстрируя использование необязательных параметров маршрута и динамическую генерацию HTML-контента.

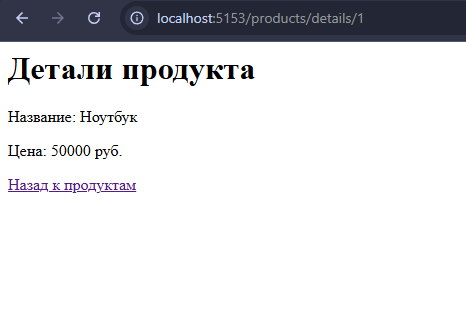
Для реализации маршрутизации применялись атрибуты [Route] и шаблоны URL, а также параметры маршрута с типами данных, что обеспечивает строгую типизацию и предотвращает ошибки при вводе некорректных значений. Использование HTML для навигации позволяет тестировать маршруты напрямую через браузер и обеспечивает визуальную наглядность работы приложения.

**Скриншоты работы приложения:**

****

****

****

****

**Листинг:**

**Program.cs:**

|  |
| --- |
| var builder = WebApplication.CreateBuilder(args);  // *Добавляем поддержку контроллеров*  builder.Services.AddControllers();  var app = builder.Build();  // *Включаем маршрутизацию контроллеров*  app.MapControllers();  // *Дополнительно можно добавить маршрут по умолчанию*  app.MapGet("/", () => "Добро пожаловать в приложение с маршрутизацией!");  app.Run(); |

**Controllers/ProductsController.cs:**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using System.Collections.Generic;  using System.Linq;  namespace Lab4.Controllers;  [Route("products")]  public class ProductController : Controller  {      // *Простейший "репозиторий" продуктов для демонстрации*      private static readonly List<Product> Products = new List<Product>      {          new Product { Id = 1, Name = "Ноутбук", Price = 50000 },          new Product { Id = 2, Name = "Смартфон", Price = 30000 },          new Product { Id = 3, Name = "Планшет", Price = 20000 }      };      [HttpGet("")]      public IActionResult Index()      {          var html = "<h1>Список продуктов</h1><ul>";          foreach (var p in Products)          {              html += $"<li><a href='/products/details/{p.Id}'>{p.Name} - {p.Price} руб.</a></li>";          }          html += "</ul><a href='/home'>На главную</a>";          return Content(html, "text/html; charset=utf-8");      }      [HttpGet("details/{id:int}")]      public IActionResult Details(int id)      {          var product = Products.FirstOrDefault(p => p.Id == id);          if (product == null)          {              return Content($"<h1>Продукт не найден</h1><a href='/products'>Назад к продуктам</a>", "text/html; charset=utf-8");          }          var html = $@"              <h1>Детали продукта</h1>              <p>Название: {product.Name}</p>              <p>Цена: {product.Price} руб.</p>              <a href='/products'>Назад к продуктам</a>          ";          return Content(html, "text/html; charset=utf-8");      }      [HttpGet("filter/{minPrice:int?}")]      public IActionResult Filter(int? minPrice)      {          var filtered = minPrice.HasValue              ? Products.Where(p => p.Price >= minPrice.Value).ToList()              : Products;          var html = "<h1>Фильтрованные продукты</h1><ul>";          foreach (var p in filtered)          {              html += $"<li>{p.Name} - {p.Price} руб.</li>";          }          html += "</ul><a href='/products'>Назад к продуктам</a>";          return Content(html, "text/html; charset=utf-8");      }  }  public class Product  {      public int Id { get; set; }      public string Name { get; set; } = string.Empty;      public int Price { get; set; }  } |

**Controllers/HomeController.cs:**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  namespace Lab4.Controllers;  [Route("home")]  public class HomeController : Controller  {      [HttpGet("")]      public IActionResult Index()      {          return Content(@"              <h1>Главная страница</h1>              <ul>                  <li><a href='/home/about'>О приложении</a></li>                  <li><a href='/home/contact'>Контакты</a></li>                  <li><a href='/products'>Продукты</a></li>              </ul>          ", "text/html; charset=utf-8");      }      [HttpGet("about")]      public IActionResult About()      {          return Content(@"              <h1>О приложении</h1>              <p>Это демонстрация маршрутизации в ASP.NET Core.</p>              <a href='/home'>Назад на главную</a>          ", "text/html; charset=utf-8");      }      [HttpGet("contact/{name?}")]      public IActionResult Contact(string? name)      {          var message = string.IsNullOrEmpty(name)              ? "Страница контактов"              : $"Страница контактов пользователя: {name}";          return Content($@"              <h1>Контакты</h1>              <p>{message}</p>              <a href='/home'>Назад на главную</a>          ", "text/html; charset=utf-8");      }  } |

**Ссылка на репозиторий:** <https://github.com/avoryyy/backend_mospoly>