**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 №1**

**Дисциплина:** BackEnd-разработка

**Тема:** Создание приложения на основе класса WebApplication в ASP.NET Core

**Выполнил: студент группы 231-332**

**Чесноков Александр Евгеньевич**

(Фамилия И.О. обучающегося)

**Дата, подпись** 01.06.2025 

(Дата)(Подпись)

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

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

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

(Дата)(Подпись)

**Москва**

**2025**

**Цель:**

Ознакомиться с базовыми шагами создания веб-приложения на основе класса WebApplication в ASP.NET Core.

**Задание:**

1. Создать новый проект ASP.NET Core приложения, используя класс WebApplication.
2. Протестировать работу приложения локально.

**Ход работы:**

Для создания простого веб-приложения на основе класса WebApplication в ASP.NET Core создан новый консольный проект с использованием .NET. В файле Program.cs была реализована минимальная структура запуска веб-приложения. Сначала с помощью WebApplication.CreateBuilder(args) был инициализирован объект билдера, затем на его основе создано само приложение через builder.Build().

После этого были добавлены три маршрута с помощью метода MapGet. Первый маршрут обрабатывает корневой адрес / и возвращает HTML-код главной страницы, содержащей заголовок и ссылки на другие страницы. Второй маршрут /about ведет на страницу с описанием проекта, где кратко поясняется, что приложение реализовано с использованием класса WebApplication. Третий маршрут /contact содержит контактную информацию и также предоставляет ссылку для возврата на главную страницу.

Каждый обработчик возвращает HTML-страницу с помощью Results.Text(...), где дополнительно указывается тип text/html; charset=utf-8 для корректного отображения кириллицы в браузере. После определения всех маршрутов приложение запускается методом app.Run(), который начинает прослушивание входящих HTTP-запросов.

**Скриншоты работы приложения:**

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

**Листинг:**

**Program.cs:**

|  |
| --- |
| public class Program  {      public static void Main(string[] args)      {          // *Создание и настройка билдера*          var builder = WebApplication.CreateBuilder(args);          var app = builder.Build();          // *Главная страница*          app.MapGet("/", () =>              Results.Text(@"<html><body>                  <h1>Главная страница лабораторной работы</h1>                  <p>Это главная страница</p>                  <a href='/about'>О проекте</a>                  <a href='/contact'>Контакты</a>              </body></html>",              "text/html; charset=utf-8")              );          // *Страница "О проекте"*          app.MapGet("/about", () =>              Results.Text(@"<html><body>                  <h1>Информация о проекте</h1>                  <p>Простое веб-приложение на ASP.NET с использованием WebApplication</p>                  <a href='/'>Главная</a>              </body></html>",              "text/html; charset=utf-8")              );          // *Страница "Контакты"*          app.MapGet("/contact", () =>              Results.Text(@"<html><body>                  <h1>Контактная информация</h1>                  <p>Код проекта представлен в листинге</p>                  <a href='/'>Главная</a>              </body></html>",              "text/html; charset=utf-8")              );          // *Запуск приложения*          app.Run();      }  } |

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