**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 №8**

**Дисциплина:** BackEnd-разработка

**Тема:** Изучение состояний в веб-приложение на основе ASP.NET Core

**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:**

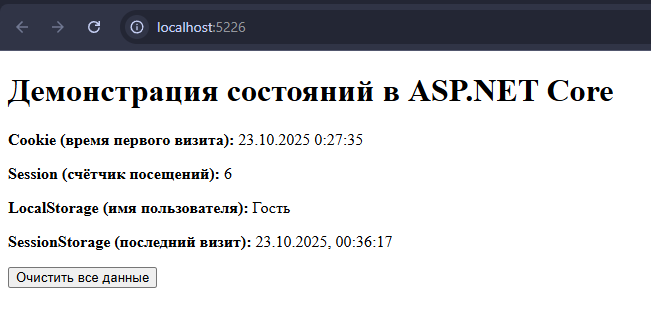
Познакомиться с основами работы с состояниями в веб-приложениях на платформе ASP.NET Core для управления информацией на клиентской и серверной сторонах.

**Ход работы:**

Первым этапом была реализована поддержка сессий. В файле Program.cs с помощью метода AddSession() были зарегистрированы необходимые сервисы, а затем добавлен middleware UseSession() для их активации в конвейере обработки запросов. Это позволило сохранять данные между обращениями одного и того же пользователя. В контроллере HomeController была добавлена логика, которая при каждом обновлении страницы увеличивает значение счётчика, хранящегося в сессии. Таким образом, пользователь мог видеть количество своих посещений сессии без необходимости хранения этой информации в базе данных или cookie.

Затем была реализована работа с **cookie-файлами**. При первом входе пользователя в систему сервер создавал cookie VisitTime, содержащую время первого визита. При последующих запросах контроллер считывал это значение из Request.Cookies, что позволило наглядно продемонстрировать сохранение данных между сеансами даже после перезапуска браузера.

Далее было добавлено использование **client-side состояний**, то есть хранилищ, доступных в браузере — localStorage и sessionStorage. Для этого в HTML-разметке страницы использовался JavaScript-код, который сохранял имя пользователя в localStorage и время последнего посещения страницы в sessionStorage. При загрузке страницы эти значения автоматически выводились в HTML, демонстрируя их доступность на стороне клиента.

**Скриншоты работы приложения:**

**Листинг:**

**Program.cs:**

|  |
| --- |
| var builder = WebApplication.CreateBuilder(args);  builder.Services.AddControllersWithViews();  builder.Services.AddSession(); // *Добавляем поддержку сессий*  builder.Services.AddEndpointsApiExplorer();  builder.Services.AddSwaggerGen();  var app = builder.Build();  app.UseStaticFiles();  app.UseSession();  if (app.Environment.IsDevelopment())  {      app.UseSwagger();      app.UseSwaggerUI();  }  app.MapControllerRoute(      name: "default",      pattern: "{controller=Home}/{action=Index}/{id?}"  );  app.Run(); |

**Controllers/HomeController.cs:**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using System.Text;  namespace Lab8.Controllers;  public class HomeController : Controller  {      public IActionResult Index()      {          // *Устанавливаем cookie, если она ещё не задана*          if (!Request.Cookies.ContainsKey("VisitTime"))          {              Response.Cookies.Append("VisitTime", DateTime.Now.ToString());          }          // *Работа с сессией*          int counter = HttpContext.Session.GetInt32("Counter") ?? 0;          counter++;          HttpContext.Session.SetInt32("Counter", counter);          var cookieValue = Request.Cookies["VisitTime"];          var sessionValue = HttpContext.Session.GetInt32("Counter");          var html = $@"          <!DOCTYPE html>          <html lang='ru'>          <head>              <meta charset='UTF-8'>              <title>Практическое занятие №8</title>              <script>                  // Работа с localStorage и sessionStorage                  if (!localStorage.getItem('username')) {{                      localStorage.setItem('username', 'Гость');                  }}                  sessionStorage.setItem('lastVisit', new Date().toLocaleString('ru-RU'));                  function showClientStorage() {{                      document.getElementById('local').innerText = localStorage.getItem('username');                      document.getElementById('session').innerText = sessionStorage.getItem('lastVisit');                  }}                  function clearClientStorage() {{                      localStorage.clear();                      sessionStorage.clear();                      alert('LocalStorage и SessionStorage очищены!');                      location.reload();                  }}              </script>          </head>          <body onload='showClientStorage()'>              <h1>Демонстрация состояний в ASP.NET Core</h1>              <p><b>Cookie (время первого визита):</b> {cookieValue}</p>              <p><b>Session (счётчик посещений):</b> {sessionValue}</p>              <p><b>LocalStorage (имя пользователя):</b> <span id='local'></span></p>              <p><b>SessionStorage (последний визит):</b> <span id='session'></span></p>              <button onclick='clearClientStorage()'>Очистить все данные</button>          </body>          </html>          ";          // *Возвращаем HTML в правильной кодировке*          Response.ContentType = "text/html; charset=utf-8";          return Content(html, "text/html", Encoding.UTF8);      }  } |

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