МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

дисциплина «Разработка приложений баз данных для информационных систем»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

«Обработка HTTP запросов средствами ASP.NET Core. Сохранение состояния. Кэширование.»

Вариант №32

Выполнил: студент гр. ИТП-31

Пикулина К. И.

Принял: доцент Асенчик О.Д.

Гомель 2022

**Цель работы:** ознакомиться c методами обработкой *HTTP* средствами *ASP*.*NET* *Core*, методами сохранения состояния приложения и повышение производительности приложений путем использования разных видов кэширования.

**Ход работы и результаты**

Используя ранее разработанные объектную модель для доступа к данным в заданной предметной области разработать простое *ASP.NET Core* приложение.

1. С использование методов Run, Map и Use разработать:
   1. компоненты промежуточного уровня (middleware) и встроить их в конвейер обработки HTTP запроса с целью кэширования 20 записей из каждой таблицы базы данных заданной предметной области с помощью встроенного инструмента кэширования - объекта IMemoryCache. Данные в кэше хранить неизменными в течение 2\*N+240 секунд, где N- номер вашего варианта.
   2. собственную систему маршрутизации входящих запросов:

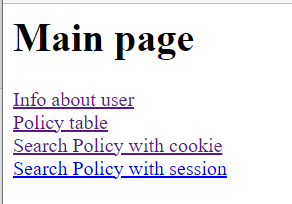


Рисунок 1 – Главная страница

* если *URL* адрес входящего запроса содержит \*info* – выводить в выходной поток для отображения браузером информацию о клиенте и выходить из конвейера обработки запроса;

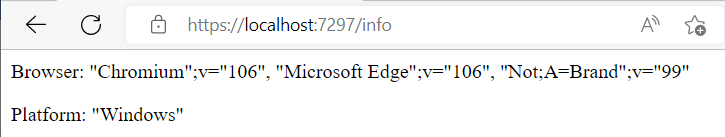


Рисунок 2 – Информация о браузере

* если *URL* адрес входящего запроса содержит \*table* (где *table* – имя таблицы из базы данных) – выводить в выходной поток для отображения браузером с использование метода *Response.WriteAsync* кэшированную информацию из соответствующей таблицы базы данных и выходить из конвейера обработки запроса;

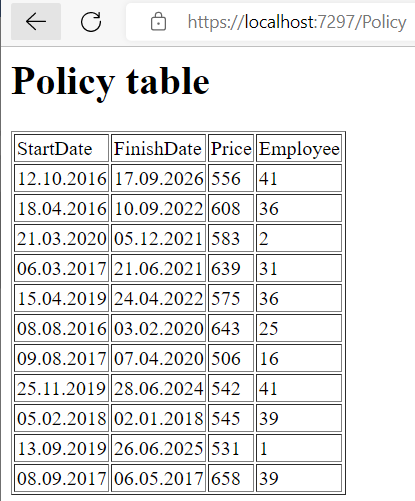


Рисунок 3 – Отображение таблицы *Policies* в браузере

* если URL адрес входящего запроса содержит \searchform1 или \searchform2 – выводить в выходной поток для отображения браузером с использование метода Response.WriteAsync формы для поиска информации из базы данных и выходить из конвейера обработки запроса;

форма должна содержать, как минимум: одно поле, одного поле со списком, один список, одну кнопку;

* в противном случае (URL адрес входящего запроса не содержит перечисленных выше элементов) - продолжать обрабатывать другие компоненты конвейера обработки запросов и передавать управление системе маршрутизации MVC фреймворка;

1. Реализовать сохранение состояния элементов одной формы одной страницы с использованием куки (\searchform1).
2. Реализовать сохранение состояния элементов одной формы одной страницы в виде одного объекта специальной структуры с использованием объекта Session (\searchform2).
3. Осуществить заполнение элементов формы при их загрузке данными ранее сохранненными в объекте Session и куки (\searchform1, \searchform2 ).



Рисунок 5 – Форма для модели Policy

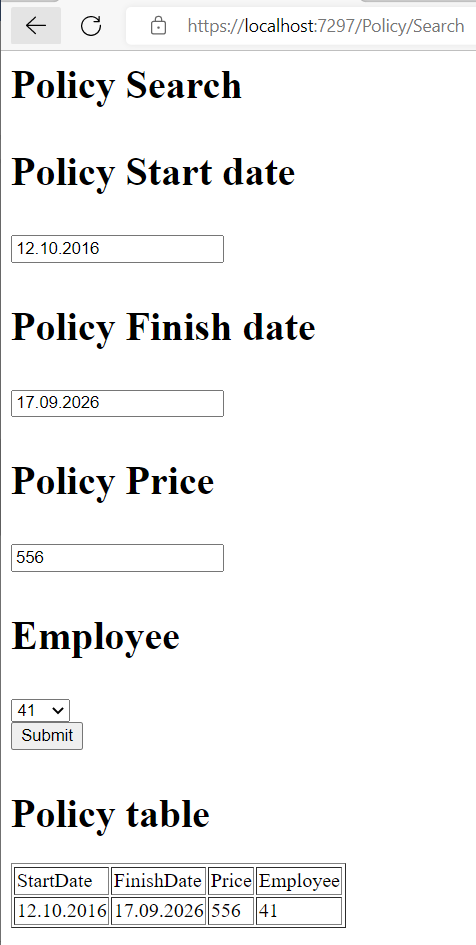


Рисунок 4 – Данные формы после перезагрузки страницы, полученные из куки



Рисунок 5 – Данные формы после перезагрузки страницы, полученные из сессии

1. С использованием средств разработчика браузера (Chrome, Firefox) продемонстрировать ускорение обработки запроса при наличии кэширования с использованием MemoryCache.

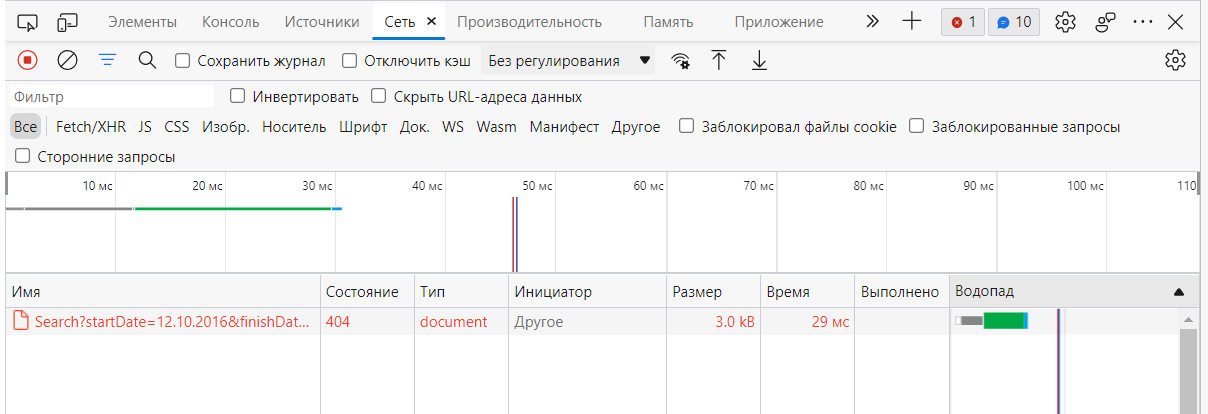


Рисунок 6 – Время с кэш

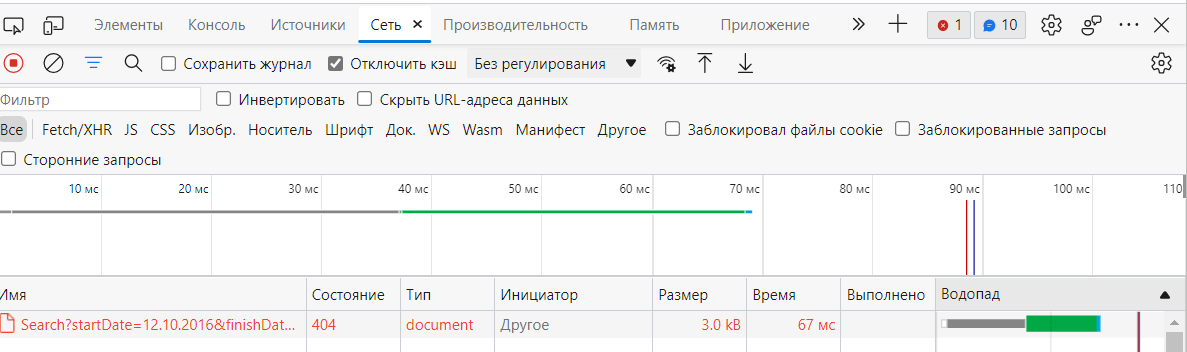


Рисунок 7 – Время без кэш

**Листинг**

[DataBaseLabs/2/3/Web at main · Ksusssa/DataBaseLabs (github.com)](https://github.com/Ksusssa/DataBaseLabs/tree/main/2/3/Web)

**Вывод:** в ходе выполнения лабораторной работы были изучены методы обработкой HTTP средствами ASP.NET Core, методы сохранения состояния приложения и повышение производительности приложений путем использования разных видов кэширования.