

УДК 004.42

MICROSOFT IDENTITY В ASP.NET CORE MVC: УПРАВЛЕНИЕ ДОСТУПОМ И СТАТИСТИКА В ЗДРАВООХРАНЕНИИ

Бийбосунов А.И., Орозобекова А.К., Бабанина И.В.

Кыргызский государственный технический университет им. И.Раззакова

Статья обсуждает создание контроллеров для программы, автоматизирующей работу администратора медицинского центра с применением технологий .Net. Основное внимание уделяется анализу преимуществ фреймворка ASP.NET Identity в обеспечении безопасности и управлении доступом администратора, а также основным технологиям разработки на платформе .Net.

Ключевые слова. Контроллеры, ASP.NET Core, клиентской части, Identity, медицинское учреждение, управление ролями.

ASP.NET CORE MVCдеги MICROSOFT IDENTITY: САЛАМАТТЫК САКТООДОГУ СТАТИСТИКА ЖАНА ЖҮКТӨЛҮҮНҮ БАШКАРУУДА МҮМКҮНДҮК АЛУУ

Бийбосунов А.И., Орозобекова А.К., Бабанина И.В.

атындагы Кыргыз мамлекеттик техникалык университети.
И.Раззакова

Макалада .Net технологияларын колдонуу менен медициналык борбордун администраторунун ишин автоматташтыруучу программа үчүн контроллерлорду түзүү талкууланат. Негизги көңүл коопсуздукту камсыз кылууда жана администратордун кирүү мүмкүнчүлүгүн башкарууда ASP.NET Identity негизинин артыкчылыктарын, ошондой эле .Net платформасында иштеп чыгуунун негизги технологияларын талдоого багытталган.

Баштапкы сөздөр. Контроллер, ASP.NET Core, кардар тарабы, Идентификация, медициналык мекеме, ролду башкаруу.

MICROSOFT IDENTITY IN ASP.NET CORE MVC: ACCESS CONTROL AND STATISTICS IN HEALTHCARE

Biybosunov A.I., Orozobekova A.K., Babanina I.V

Kyrgyz State Technical University named after. I.Razzakova

The article discusses the creation of controllers for a program that automates the work of a medical center administrator using .Net technologies. The main focus is on analyzing the benefits of the ASP.NET Identity framework in providing security and managing administrator access, as well as the main development technologies on the .Net platform.

Keywords. Controllers, ASP.NET Core, client side, Identity, medical facility, role management.

С расширением сферы компьютерных технологий на предприятии возникла потребность в разработке программного продукта, направленного на оптимизацию человеческих трудозатрат и повышение оперативности выдачи результатов, необходимых для работы сотрудников. Ключевой задачей такого программного продукта является замена традиционных методов хранения информации в форме больших архивов на структурированный электронный формат, что способствует эффективному управлению данными и обеспечивает более эффективный доступ к необходимой информации. Для данной системы важным фактором являются технические характеристики оборудования (объемы и скорости накопителей информации, типы процессоров), а также эффективно организованная база данных, так и сам пользовательский интерфейс.

Центральным звеном в архитектуре ASP.NET Core MVC является контроллер. При получении запроса система маршрутизации выбирает для обработки запроса нужный контроллер и передает ему данные запроса. Контроллер обрабатывает эти данные и посылает обратно результат обработки.

В ASP.NET Core MVC контроллер представляет собой класс веб-приложения на языке C#, который наследуется от абстрактного базового класса `Microsoft.AspNetCore.Mvc.Controller`. По умолчанию проект ASP.NET Core MVC содержит как минимум один контроллер - `HomeController`:

При использовании контроллеров существуют некоторые условности. Во-первых, в проекте контроллеры помещаются в каталог `Controllers`. И во-вторых, по соглашениям об именовании названия контроллеров обычно оканчиваются на суффикс `"Controller"`, либо наследовать класса `"Controller"` или `"ControllerBase"`, и могут храниться где угодно а не именно в папке `"Controllers"`. Компилятор сам узнает что чего есть.

Контроллер, как и любой класс на языке `C#`, может иметь поля, свойства, методы. По умолчанию `HomeController` имеет четыре метода, которые можно назвать действиями. Действия контроллера - это публичные методы, которые могут сопоставляться с запросами. Например, стандартный контроллер содержит метод `Index` - он имеет модификатор `public` и поэтому может использоваться для обработки запроса.

`ASP.NET Core` — это модификация `ASP.NET 4.x` с архитектурными изменениями, формирующими более рациональную и более модульную платформу.

`ASP.NET Core` предоставляет следующие преимущества:

- Единое решение для создания пользовательского веб-интерфейса и веб-API.
- Разработано для тестируемости.
- `Razor Pages` упрощает написание кода для сценариев страниц и повышает его эффективность.
- `Blazor` позволяет использовать в браузере язык `C#` вместе с `JavaScript`. совместное использование серверной и клиентской логик приложений, написанных с помощью `.NET`;
- Возможность разработки и запуска в ОС `Windows`, `macOS` и `Linux`.
- Открытый исходный код и ориентация на сообщество.
- Интеграция современных клиентских платформ и рабочих процессов разработки.

- Поддержка размещения служб удаленного вызова процедур (RPC) с помощью gRPC.
- Облачная система конфигурации на основе среды.
- Встроенное введение зависимостей.
- Упрощенный высокопроизводительный модульный конвейер HTTP-запросов.

Внедрение информационной системы учета на в медицинском центре подразумевает собой создание базы данных, выполняющей следующие функции:

- ввод и просмотр входной информации;
- добавление, удаление и корректировка информации;
- обработка входной информации;
- получение оперативной информации по учету пациентов;
- формирование выходных документов.

Процесс разработки осуществляется в интегрированной среде разработки Microsoft Visual Studio 2022, на рисунке (Рис.1.) представлен интерфейс создания приложения в Microsoft Visual Studio 2022. Здесь разработчик может определить основные параметры проекта, такие как тип приложения, используемые технологии, настройки безопасности и другие параметры

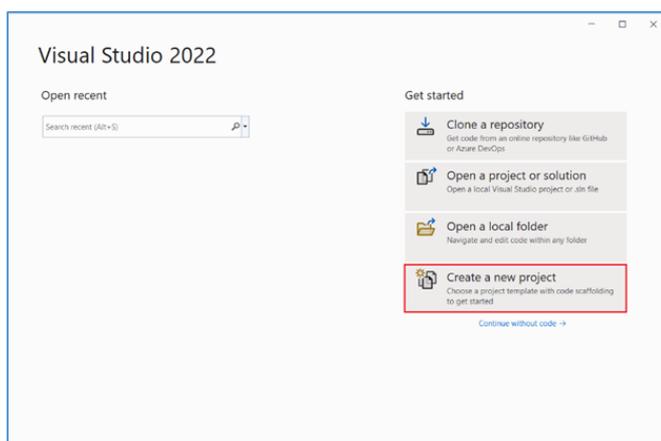
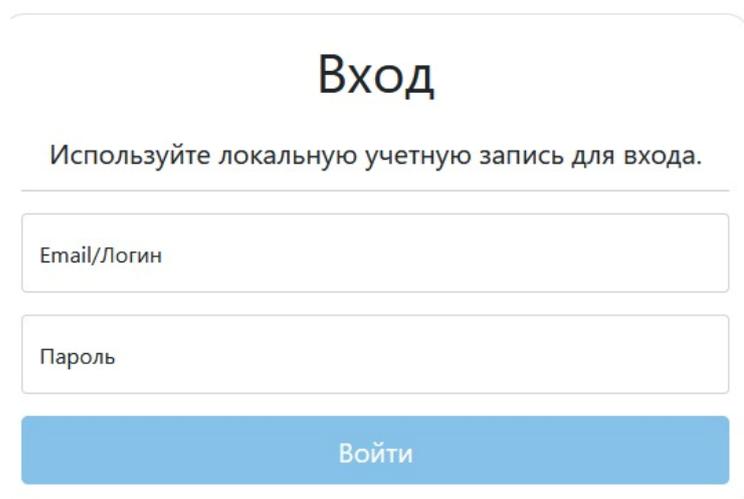


Рис.1. Запуск Visual Studio 2022 и создание проекта

Для реализации безопасности и управления учетными данными в клиентской части приложения внедрен Microsoft Identity в ASP.NET Core, предоставляющий следующие возможности:

- Microsoft Identity используется для обеспечения аутентификации и авторизации пользователей. Этот интегрированный инструментариий обеспечивает безопасный вход в систему, управление ролями и политиками доступа, а также поддержку внешних провайдеров аутентификации.
- Microsoft Identity в ASP.NET Core включает в себя встроенные механизмы защиты от атак CSRF, что повышает безопасность клиентской части приложения и предотвращает поддельные запросы от злоумышленников.
- Microsoft Identity позволяет настраивать пользовательские атрибуты и профили, что обеспечивает гибкость в управлении данными о пользователях и их персонализацию. Возможность внедрения многофакторной аутентификации для повышения уровня безопасности, требуя дополнительных шагов для подтверждения личности пользователя.

На рисунке (Рис.2) ниже представлена страница входа, созданная с помощью Microsoft Identity, разработанной информационной системы.



Вход

Используйте локальную учетную запись для входа.

Email/Логин

Пароль

Войти

Рис. 2. Форма аторизации в приложение

Microsoft Identity в ASP.NET Core обеспечивает мощное управление ролями для учетных записей в приложении. Он предоставляет следующие ключевые возможности:

Ролевая аутентификация: Microsoft Identity позволяет присваивать роли пользователям, определяя их права и доступ к определенным частям приложения.

- Авторизация на уровне ролей: Роли используются для определения, какие части приложения могут быть доступны для определенных пользовательских групп.
- Программное управление ролями: Разработчики могут динамически управлять ролями пользователей, включая назначение и удаление ролей.
- Пользовательские роли: Помимо встроенных ролей, можно создавать пользовательские роли в соответствии с требованиями приложения.
- Применение управления ролями в Microsoft Identity обеспечивает эффективную организацию доступа пользователей к функционалу приложения в соответствии с их ролями и обязанностями.

Страница управления (Рис.3) ролями предоставляет функционал по администрированию ролей в приложении, включая добавление, удаление, блокирование и изменение пользователей и их ролей. Всего в приложении определены и добавлены 5 ролей супер администратор, администратор, врач, главный врач, заведующий отделением и обычный пользователь, используя подход Code-First для создания и управления базой данных. Стоит отметить что добавлять новых пользователей а так же редактировать их данные может только администратор.

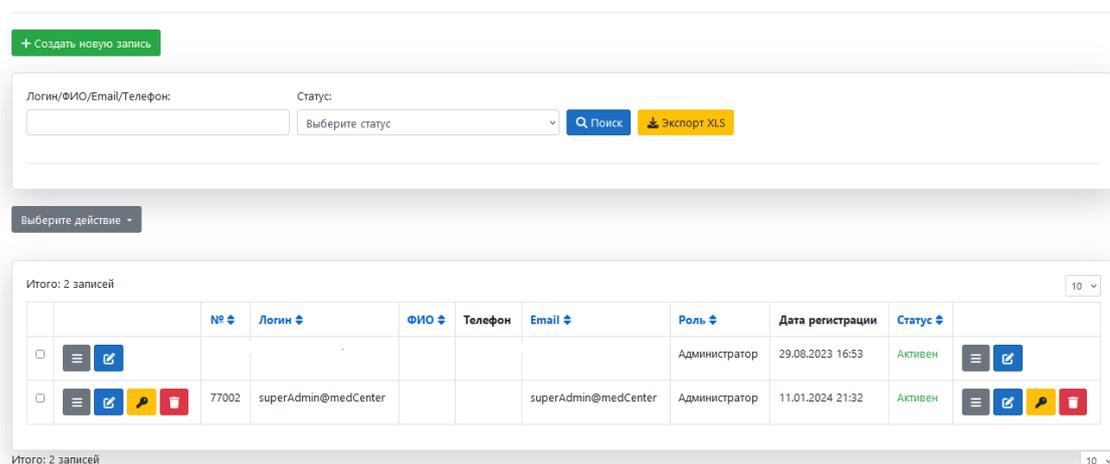


Рис.3. Главная страница управления пользователями

После успешной аутентификации в приложении администратор направляется на главную страницу (Рис.4), на которой предоставлен разносторонний функционал, включая, но не ограничиваясь: возможность записи на прием, проверку текущих записей, интерактивный календарь событий и другие важные элементы управления. Помимо основных компонентов на странице, доступных для выполнения операций, реализовано верхнее меню, в котором администратор может обнаружить и получить доступ ко всем остальным элементам и функциям приложения. Это верхнее меню служит навигационной платформой, обеспечивая удобное распределение функциональных возможностей приложения и обеспечивая оперативный доступ к необходимым разделам и инструментам.

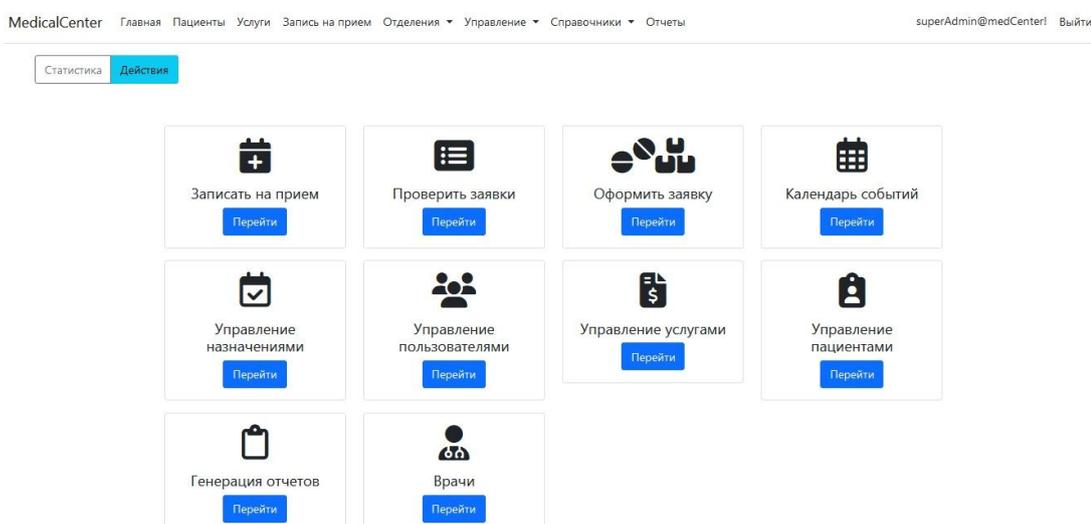


Рис.4. Главная страница приложения

На главной странице приложения предоставлена не только основная функциональность, но и вкладка "Статистика", которая представляет собой контекст аналитической информации (Рис.5), специфически связанной с медицинским центром. На данной вкладке пользователь может получить обобщенное представление о текущем состоянии и эффективности работы отделений медицинского учреждения.

Информация в разделе "Статистика" представлена в виде карточек, содержащих ключевые метрики по каждому отделению:

- Отделение и ссылка: Указание на конкретное отделение, а также возможность перейти к детальной информации, представленной в виде ссылки. Это обеспечивает навигацию пользователя по статистическим данным для более подробного анализа.
- Количество палат: Общее число палат в каждом отделении является важным показателем структуры медицинского учреждения.
- Количество коек: Информация о общем количестве койко-мест в каждом отделении.
- Занятые и свободные места: Отображение актуального количества занятых и свободных койко-мест, что позволяет оперативно оценивать загрузку каждого отделения.

Информация о количестве пациентов: Позволяет быстро оценить количество пациентов, находящихся в данном отделении, что является важным параметром для общего контроля и планирования.

Эти статистические данные на странице "Статистика" обеспечивают администратора системы необходимой информацией для оперативного принятия управленческих решений и обеспечивают более эффективное управление ресурсами медицинского учреждения.

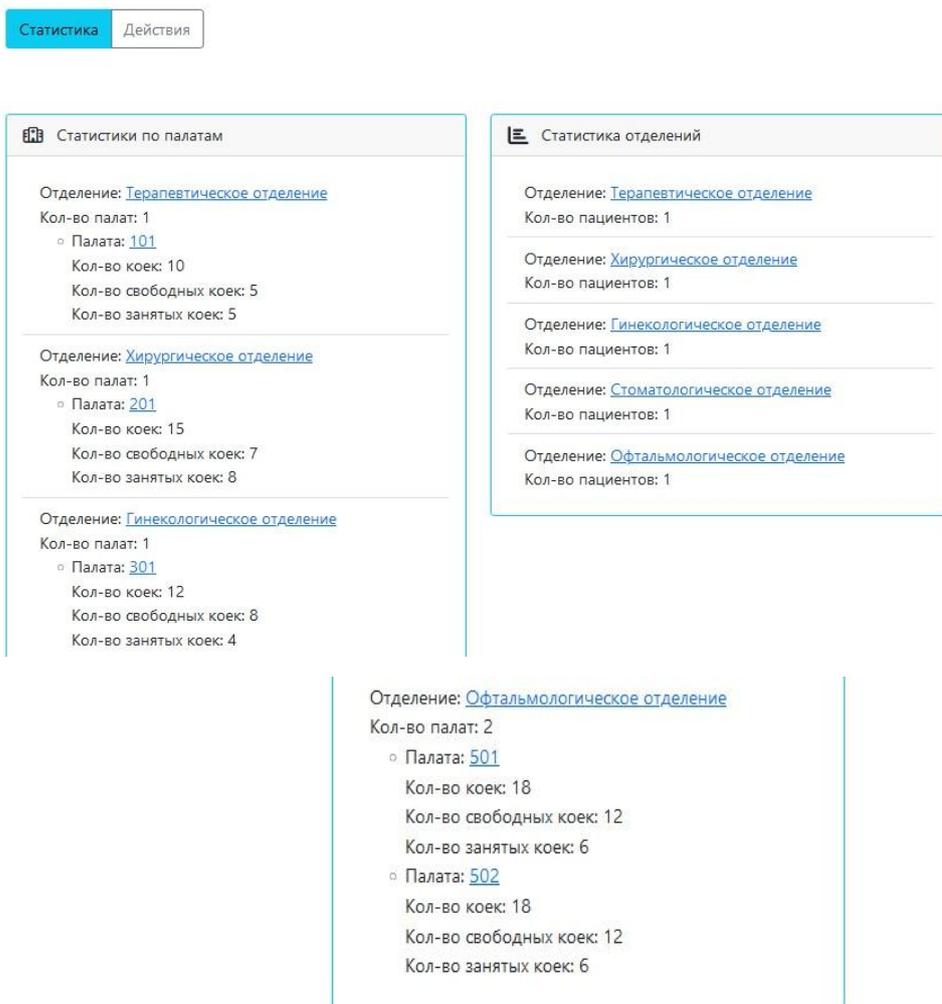


Рис.5. Раздел "Статистика"

Разработка клиентской части включает Bootstrap 5 для адаптивного дизайна, jQuery и JavaScript для динамического поведения, и Font Awesome для улучшения визуального опыта. Для асинхронного взаимодействия с сервером используется AJAX.

Серверная часть, основанная на .NET Core 7, включает ASP.NET MVC и Web API для создания масштабируемого приложения. В работе с базой данных используется Entity Framework с подходом Code-First. Технология LINQ применяется для удобства работы с данными в коде.

MSSQL Studio 2018 и SQL Server 2022 выбраны в качестве систем управления базами данных, а Microsoft Visual Studio 2022 Community — для разработки и отладки.

Заключение. Разработанная информационная система для медицинского учреждения, основанная на платформе ASP.NET Core MVC и интегрированная с Microsoft Identity, продемонстрировала высокую эффективность в автоматизации учета и обработки медицинской информации. Система обеспечивает высокий уровень безопасности и гибкость управления доступом, что содействует эффективному взаимодействию персонала и повышает общую производительность медицинского учреждения. Полученные результаты подтверждают успешность реализации проекта.

ЛИТЕРАТУРА

1. <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio>
2. <https://habr.com/ru/companies/microsoft/articles/245901/>
3. <https://habr.com/ru/articles/672382/>
4. Кристиан Венц. ASP.NET Core Security 2022г
5. Jon P Smith. Entity Framework Core in Action, Second Edition 2021г.
6. Эндрю Лок ASP.NET Core in Action, Third Edition 2023г.
7. Радионов, Р. А. Анализ методов оперативного контроля на предприятии за фактическим состоянием запасов. 2014 г.