МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета – Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

Отчет по курсовому проекту

ПМ 01. Разработка программных модулей

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ “MYDUOCARDS”**

| Руководитель | Волховицкий Г.В.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023 г. |
| --- | --- |
| Студент 3 курса  гр. 107 г2 | Половых Н.А.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2023 г. |

Новосибирск

2023

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ](#_3x8tuzt) 3

[ВВЕДЕНИЕ](#_2ce457m) 4

[1 ПОСТАНОВКА ЗАДАЧИ ВКР](#_2et92p0) 5

[1.1 Бизнес-требования](#_rjefff) 5

[1.2 Пользовательские требования](#_1t3h5sf) 5

[1.3 Системные требования](#_2s8eyo1) 5

[1.4 Требования к графическому пользовательскому интерфейсу](#_17dp8vu) 6

[1.5 План-график выполнения ВКР](#_3rdcrjn) 6

[2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ](#_3bj1y38) 7

[2.1 Описание предметной области задачи ВКР](#_3as4poj)7

[2.2 Классы и характеристики пользователей](#_ihv636) 7

[2.3 Функциональные требования](#_vx1227)8

[2.4 Нефункциональные требования](#_3fwokq0) 8

[3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ](#_2pta16n) 10

[3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки](#_14ykbeg) 10

[3.2 Характеристика выбранных программных сред и средств 1](#_4bvk7pj)2

[4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 1](#_2afmg28)3

[4.1 Этапы реализации ПС (ПМ) 1](#_1opuj5n)3

[4.2 Пользовательский интерфейс ПС (ПМ)](#_1opuj5n) 14

[4.3 Входные, выходные и промежуточные данные](#_3oy7u29) 16

[4.4](#_1opuj5n) [Разработка базы данных, реализуемой в рамках ПС (ПМ)](#_184mhaj) 16

[4.5 Архитектура и схема функционирования ПС (ПМ)](#_184mhaj)18

[5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯ](#_j8sehv) 20

[5.1 План тестирования](#_279ka65) 20

[5.2 Результаты тестирования](#_meukdy) 20

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ](#_j8sehv) 22

[6.1 Руководство для обычного пользователя](#_279ka65) 22

[6.2 Руководство для администратора](#_279ka65) 25

[ЗАКЛЮЧЕНИЕ](#_1ljsd9k) 28

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ](#_338fx5o) 29

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

1. ASP.NET Core: Свободно-распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET.
2. MVC: Model-View-Controller - паттерн проектирования, который разделяет приложение на три компонента: Model (модель), View (представление) и Controller (контроллер).
3. Хостинг - услуга по предоставлению ресурсов для размещения информации на сервере, постоянно имеющем доступ к сети.
4. Интерфейс - совокупность средств, обеспечивающих взаимодействие функциональных устройств и/или программ в вычислительной системе (компьютере), а также взаимодействие их с пользователем.
5. Репозиторий - место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.
6. Фре́ймворк; иногда фреймво́рк - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
7. Bootstrap - свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.
8. Git - распределённая система управления версиями.
9. ПО - “Программное обеспечение”.
10. CRUD - акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление.
11. Layout - макет оформления веб-страницы (изначально книг, газет).

ВВЕДЕНИЕ

Данный курсовой проект посвящен разработке веб-приложения "MyDuoCards", предназначенного для обучения языкам. Актуальность данной работы обусловлена постоянным ростом интереса к изучению языков различных стран и культур, а также необходимостью предоставления доступных, удобных и инновационных средств для этой цели.

В контексте современных технологий и практик обучения, разработка "MyDuoCards" имеет целью предложить эффективное и интуитивно понятное веб-приложение, способствующее улучшению процесса изучения языков и созданию поддерживающей и стимулирующей среды пользователей. Разработка такого приложения представляет значимый интерес для обучающих учреждений, студентов, преподавателей и всех желающих освоить новый язык в удобной и доступной форме.

Подходящее и инновационное веб-приложение может стать ценным инструментом в современном мире, где неотъемлемую часть жизни каждого человека занимает постоянно развивающаяся и растущая сфера цифровых технологий.

1 ПОСТАНОВКА ЗАДАЧИ ВКР

Суть решаемой задачи заключается в разработке и реализации веб-приложения "MyDuoCards", предназначенного для обучения языкам. Проект включает в себя создание удобного, интуитивно понятного интерфейса, а также функционала, который поддерживает эффективные методы с правильно прописанной реализацией ради удобства на как на уровне использования приложения, так и на уровне качества запоминаемой информации пользователем.

**1.1 Бизнес-требования**

* Общее целевое назначение разрабатываемого программного средства - предоставить пользователям удобный и эффективный инструмент для изучения иностранных языков.
* Данное приложение предназначено для широкого спектра пользователей, желающих изучать различные иностранные языки с помощью интерактивных заданий и карточек.

**1.2 Пользовательские требования**

* Пользователи получат возможность изучать выбранные языки через выполнение интерактивных упражнений и заданий, а также отслеживать свой прогресс в изучении языков.
* Разрабатываемое ПС предназначено для автономной работы и не является частью более крупной системы.

**1.3 Системные требования**

Разрабатываемое приложение "MyDuoCards" будет состоять из проекта ASP.NET MVC и реляционной базы данных, хранящей в себе всю требуемую для ПС информацию.

**1.4 Требования к графическому пользовательскому интерфейсу**

Пользовательский интерфейс ПС должен обладать интуитивной навигацией, привлекательным дизайном и обеспечивать удобные способы взаимодействия пользователя с приложением, создающие комфортную среду для непосредственного изучения иностранного языков.

**1.5 План-график выполнения ВКР**

1. Первая неделя:
   1. Изучение существующих аналогов и анализ их функциональности.
   2. Определение основных этапов разработки и составление плана работы.
2. Вторая неделя:
   1. Разработка логики аутентификации и управления учетной записью пользователя.
   2. Тестирование и отладка нововведений.
3. Третья неделя:
   1. Создание отображения карточек пользователя и словаря для всего приложения.
   2. Проверка реализации созданных функций и их доработка.
4. Четвертая неделя:
   1. Написание и внедрение возможности поиска на страницах и переключения между станицами с карточками.
   2. Проверка работоспособности логики в системы и исправление ошибок.
5. Пятая неделя:
   1. Финальное тестирование и отладка приложения.
   2. Подготовка к защите ВКР.

2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ

**2.1 Описание предметной области задачи ВКР**

Предметной областью приложения "MyDuoCards" является обучение иностранным языкам. Разработка такого приложения предполагает углубленное понимание языкового обучения и принципов эффективной коммуникации.

В предметной области "MyDuoCards" важными аспектами являются:

* Методики обучения: Изучение различных методик и подходов в обучении иностранным языкам, включая интерактивные методики, контекстное обучение и адаптивные подходы.
* Персонализация обучения: Эффективное адаптирование программы обучения к индивидуальным потребностям и уровню знаний каждого пользователя.
* Эффективное измерение прогресса: Отслеживание и анализ прогресса пользователей, чтобы обеспечить адаптацию программы обучения и мотивацию для пользователей.
* Исследование и понимание указанных аспектов предметной области помогут разработать эффективное и инновационное приложение "MyDuoCards", способствующее усвоению иностранных языков пользователями.

**2.2 Классы и характеристики пользователей**

Пользователи приложения "MyDuoCards" могут быть разделены на несколько классов в зависимости от их потребностей, целей и предпочтений. Ниже представлены основные классы пользователей и их характеристики:

* Обычный пользователь: Это простые пользователи приложения, возможности которых заканчиваются на логике обучения. Этим пользователям нет доступа для редактирования общего словаря, а также редактирования записей в любой из сущностей в базе данных.

* Администратор: Пользователь, имеющий дополнительные возможности относительно обычного, которые заключаются в возможности редактирования некоторых аспектов поведения веб-приложения через доступный только администраторам интерфейс.

Каждый класс пользователей имеет свои уникальные потребности в обучении, и разработка "MyDuoCards" должна учитывать эти различия, предлагая персонализированные курсы, материалы и методики обучения, чтобы обеспечить эффективное и удовлетворительное обучение для каждого пользователя.

**2.3 Функциональные требования**

Функциональные требования к приложению "MyDuoCards" заключаются в основополагающих вещах для осмысленной работы приложения и включают в себя такие пункты как:

* Регистрация/аутентификация;
* Персонализированные профили;
* Создание карточек для запоминания;
* Интерактивные упражнения;
* Отслеживание прогресса;
* Поддержку различных языков;
* Административные функции и доступность на различных устройствах.

**2.4 Нефункциональные требования**

Разрабатываемое ПО также имеет нижеизложенные качества, пусть периодически в крайне упрощенном варианте их реализации вследствие неопытности разработков:

1. Производительность и масштабируемость

Достигаются использованием оптимизированных алгоритмов серверной логики.

2. Безопасность

В приложении используется ValidateAntiforgeryToken предназначенный для противодействия подделке межсайтовых запросов, производя верификацию токенов при обращении к методу действия.

3. Удобство использования

Разрабатываемое ПО предусматривает обширное ветвление переходов по страницам и, в нужных местах, предусматривает ограничения.

4. Сопровождаемость: Наличие требования к простоте сопровождения, управления конфигурацией, журналированию и мониторингу.

3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ

Разрабатываемое веб-приложение не имеет большого количества модулей. Выше было описано, что оно будет представлять из себя проект ASP.NET MVC и сопряжённую с ним базу данных, а значит следует проанализировать и выбрать подходящие, под эти два модуля, средства.

**3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки**

Первой рассмотри средство разработки само проекта. Выбранная среда должна быть проста в использовании в угоду выполнения работы за минимальные сроки и иметь широкий функционал возможностей при разработке.

**“Visual Studio”:**

Достоинства:

* Интегрированная среда разработки (IDE) с обширным набором инструментов для создания приложений ASP.NET MVC.
* Полноценная поддержка разработки .NET Framework и .NET Core, включая инструменты для создания, отладки и развертывания ASP.NET приложений.
* Интеграция с множеством расширений и инструментов сторонних разработчиков.

Недостатки:

* Требует большого количества ресурсов, особенно в случае больших проектов.

**“Visual Studio Code”:**

Достоинства:

* Легковесный и быстрый редактор кода с обширной поддержкой языков программирования и расширений.
* Поддержка интеграции с Git, отладчиками и расширениями для работы с ASP.NET MVC проектами.
* Широкий выбор расширений, позволяющих настраивать среду разработки под конкретные потребности.

Недостатки:

* Не столь обширные инструменты для разработки как у полноценной среды Visual Studio.
* Возможно потребуется установка дополнительных расширений и настройка среды для полноценной поддержки ASP.NET MVC.

Далее следует выбрать базу данных, с которой будет просто взаимодействовать с разных рабочих мест при отсутствии хостинга, которая не будет требовательна по системным требованиям.

**“SQLite”:**

Достоинства:

* Легко встраиваемая база данных, не требующая отдельного сервера для работы. Это означает, что вы можете работать с SQLite с разных рабочих мест, не настраивая специальных серверов.
* Прост в использовании и настройке, что упрощает рабочий процесс и особенно удобен для небольших проектов.
* Отлично подходит для небольших приложений с локальным хранением данных.

Недостатки:

* Может быть менее эффективным при работе с большими объемами данных и при одновременном доступе нескольких пользователей по сравнению с серверными базами данных, такими как MS SQL.
* Ограниченные возможности масштабирования и распределенных вычислений.

**“MS SQL Server”:**

Достоинства:

* Поддерживает масштабирование, эффективно работает с большими объемами данных и обеспечивает возможность одновременного доступа нескольких пользователей.
* Обеспечивает широкий набор функций, процедур, триггеров, индексов, что дает больше возможностей для разработки сложных приложений.
* Поддерживает удаленный доступ из различных рабочих мест без необходимости локальной установки базы данных.

Недостатки:

* Требует развертывания и настройки серверной части, что может потребовать больше времени и уровень компетенций для настройки и обслуживания.
* Больше затрат на лицензии и инфраструктуру по сравнению с SQLite.

**3.2 Характеристика выбранных программных сред и средств**

Ввиду строгих временных рамок и действительно существенных минусов Visual Studio Code, выбор пал в сторону Visual Studio при разработке веб-приложения.

С выбором базы данных возникли трудности ввиду отсутствия хостинга для её размещения, вследствие чего выбор остановился на использования простой и портативной базы данных SQLite.

Также для разработки ПС использовалась система контроля версий Git. Аккаунт для неё был выделен научным руководителем, из-за чего альтернативы не рассматривались в принципе.

4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

**4.1 Этапы реализации ПС (ПМ)**

1. Анализ требований:

Провести исследование рынка образовательных приложений для изучения подходов, технологий и лучших практик.

Определить основные функциональные и нефункциональные требования к приложению.

1. Проектирование системы:

Разработать архитектуру приложения, включая базу данных, бэкэнд и фронтенд компоненты.

Создать диаграмму базы данных для хранения информации о карточках, пользовательских данных и статистике обучения.

1. Разработка бэкэнда:

Создание серверной части приложения с использованием фреймворка ASP.NET Core.

1. Разработка фронтенда:

Создание пользовательского интерфейса с помощью HTML, CSS и JavaScript.

Внедрение функционала изучения карточек, статистики прогресса, а также редактирования и создания новых карточек.

1. Интеграция базы данных:

Настройка и подключение базы данных для хранения информации о карточках, пользовательских аккаунтах и их прогрессе.

1. Тестирование:

Проведение модульного, интеграционного и приемочного тестирования для обеспечения качества кода и функциональности приложения.

**4.2 Пользовательский интерфейс ПС (ПМ)** Интерфейс приложения не представляет из себя что-то новое или новаторское, однако он разрабатывался с учётом практичности и удобства в руках любого пользователя. Создание интерфейса предполагало размещение всех важных и неважных элементов на местах их значимости, тогда как архитектура html-страницы основана на общепринятых для разработки стандартах создания веб-страниц.

Если же рассматривать всё приложение постранично, то у нас получается следующая карта страниц:

1. Входом в веб-приложение выступает страница авторизации. Её индивидуальная логика - авторизация или же направление пользователя на окно регистрации в приложении.
2. Вышеупомянутое окно регистрации является страницей с формой, которая отвечает за запись новых пользователей в базу данных для последующего использования приложением. Данное окно имеет переход лишь на страницу авторизации.
3. При входе в приложение пользователя переносит на страницу с именем “Home”, которая выступает непосредственно страницей обучения. Здесь отображаются карточки, добавленные пользователем, его логин, кнопка выхода из аккаунта и последующего переноса пользователя на страницу входа, форма для фильтрации карточек и, последнее, кнопки, позволяющие увидеть все карточка пользователя, так как при их большом количестве, на одном экране они не уместятся.
4. Страница “Vocabulary” выступает словарём всего приложения. Она очень схожа с предыдущей страницей по наполнению, однако информации о каком-либо пользователе или же его конкретных карточек тут нет.
5. Страница “Options” заключает в себя форму, которая позволяет пользователю обновлять информацию о себе в базе данных.

Следует отметить, что страницы “Home”, “Options” и “Vocabulary” недоступны незарегистрированным пользователям. Даже прописав конкретный редирект, пользователь без авторизации не сможет на них перейти.

В приложение также присутствуют недоступные обычным пользователям страницы. Данные страницы, а если быть точнее, группы страниц доступны только пользователям с ролью “Администратор” и описывают CRUD-функции вокруг конкретной сущности в базе данных. Каждый такой кластер состоит из:

1. Страница “Index”. Здесь высвечивается полный список записей в базе данных по определенной модели, однако так же, как и у карточек на странице “Home” или же “Vocabulary” здесь присутствует поиск и есть кнопки пагинации. Также с этой страницы осуществляется переход на все нижеописанные с помощью кнопок у каждой записи за исключением страницы добавления записей.
2. Страница “Details”. Здесь описывается конкретно выбранная запись сущности и присутствует переход на страницу “Edit” и кнопка перехода к странице “Index”.
3. Страница “Edit”. Данная страница позволяет редактировать конкретную запись. Здесь присутствует кнопка перехода к странице “Index”.
4. Страница “Delete”. Эта страница позволяет удалить конкретную запись. Здесь также присутствует кнопка перехода к странице “Index”.
5. Страница “Create”. Эта страницы не определяется к конкретной записи у сущности в базе данных, напротив ,она имеет логику добавления новых записей к сущности. На ней имеется только обратный переход ко списку элементов страницы “Index”.

Помимо всего прочего, большая часть логики навигации располагается на частичной странице layout, которая выступает, условно, основой для каждой из страниц. Layout обладает различными переходами в зависимости от авторизации и роли пользователя.

**4.3 Входные, выходные и промежуточные данные**

Входными и одновременно выходными данными у проекта можно обозначить только изменения файлов базы данных по ходу использования веб-приложения.

**4.4 Разработка базы данных, реализуемой в рамках ПС (ПМ)**

В данном пункте описываются конкретные сущности базы данных и их всевозможные характеристики.

1. Таблица: Attendances

Назначение: Хранение записей с активностью пользователей.

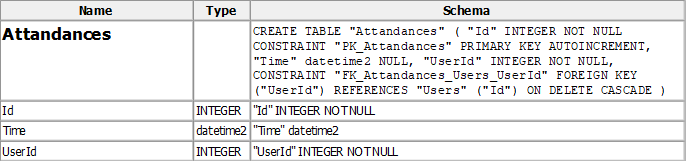


Рисунок - 1 Attendances

1. Таблица: Dictionaries

Назначение: Хранение “карточек”, которые добавляют пользователи к себе.

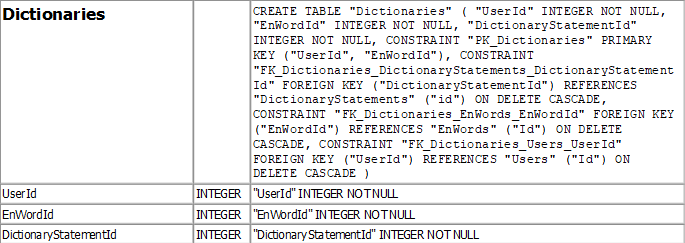


Рисунок - 2 Dictionaries

1. Таблица: DictionaryStatements

Назначение: Хранение статусов для каждой карточки, а соответственно, ее приоритет для обучения у пользователя.

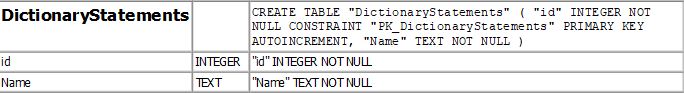


Рисунок - 3 DictionaryStatements

1. Таблица: EnWords

Назначение: Хранение английских слов, основная таблица для языковой логики.

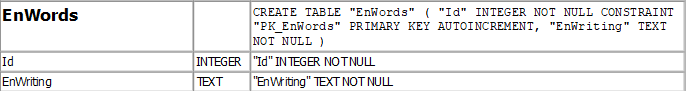


Рисунок - 4 EnWords

1. Таблица: Roles

Назначение: Хранение ролей пользователей, относительно которых зависит доступ к различной логике приложения.

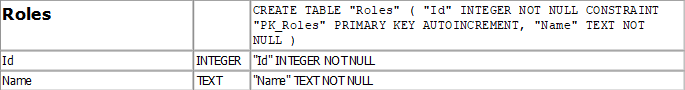


Рисунок - 5 Roles

1. Таблица: RuWords

Назначение: Хранение переводов английских слов на русский язык.

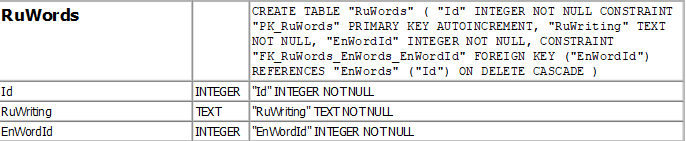


Рисунок - 6 RuWords

1. Таблица: Users

Назначение: Хранение непосредственных данных пользователей.

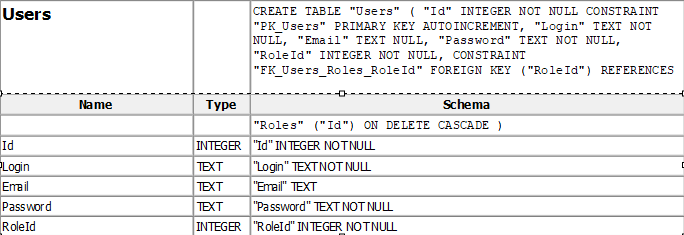


Рисунок - 7 Users

**4.5 Архитектура и схема функционирования ПС (ПМ)**

В приложении присутствует всего десять контроллеров. Некоторые отвечают за логику конкретных веб-страниц, какие-то же имеют более гибкое значение для приложения.

Первые шесть контроллеров являются CRUD-контроллерами и отвечают за CRUD-логику у своей определённой сущности. По сути это в чём-то доработанные изначальные контроллеры определённой модели Entity Framework.

Следующие четыре контроллера имеют более произвольные ответственности:

1. “HomeController”. Полностью отвечает за страницу “Home”. Также содержит внутри себя логику отслеживания и отображения посещения приложения пользователями.

2. “VocabularyController”. Данный контроллер, как и HomeController отвечает за страницу “Vocabulary”. Этот контроллер содержит внутри себя функционал добавления и удаления карточек. В свою очередь, данные методы присутствуют на “Home” и “Vocabulary” страницах и являются важным аспектом приложения.

3. “OptionsController” - контроллер, описывающий поведение страницы “Options”, на которой пользователь имеет возможность изменить настройки своего профиля.

4. “AccountController”. Последний контроллер полностью вобрал в себя реализацию регистрации, авторизации и деавторизации пользователя. Он отвечает за такие страницы как “Login” и “Register”.

5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯ

**5.1 План тестирования**

План тестирования заключается в проверке приложения на работоспособность после каждого добавленного нововведения. Тестирование происходит сразу же ради скорейшего выявления конкретной проблемы внутри добавленного контента.

Если результаты тестирования не удовлетворяют функциональным требованиям, добавленная логика уходит на доработку или же оптимизацию и не заносится на репозиторий. В исключительных случаях исправление ошибок откладывается на небольшой срок в угоду качественного разрешения функциональных проблем, а не преждевременных поисков компромиссов.

**5.2 Результаты тестирования**

Результаты описаны с опорой на план-график написания ВКР.

1. Тестирование аналогов:

Результат: Проведено исследование существующих аналогов приложения, что позволило составить подробный отчет о их функциональности и особенностях.

1. Тестирование логики аутентификации:

Результат: Процесс аутентификации успешно протестирован для различных сценариев, демонстрирующих его безопасность и надежность. Это гарантирует безопасность пользователей при входе в систему.

1. Тестирование отображения карточек пользователя и словаря:

Результат: Успешно проведена проверка отображения пользовательских карточек на различных устройствах, а также удостоверено, что словарь содержит необходимую информацию. Это гарантирует удобство использования приложения для пользователей.

1. Тестирование функции поиска и переключения между страницами с карточками:

Результат: Функция поиска успешно протестирована по различным параметрам, а также отмечено корректное переключение между страницами с карточками, что подтверждает ее функциональность и удобство использования.

1. Финальное тестирование и отладка:

Результат: В ходе финального тестирования обнаружены и зафиксированы все обнаруженные ошибки и недочеты, которые успешно устранены, обеспечивая стабильную работу приложения.

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В данном разделе описывается последовательность действий под определённую роль для комфортного использования веб-приложения “MyDuoCards”.

**6.1 Руководство для обычного пользователя**

* Авторизация и регистрация:

При открытии приложения вас перенаправит на страницу авторизации. Если у вас нет учетной записи, перейдите на страницу регистрации, заполните форму и зарегистрируйтесь.

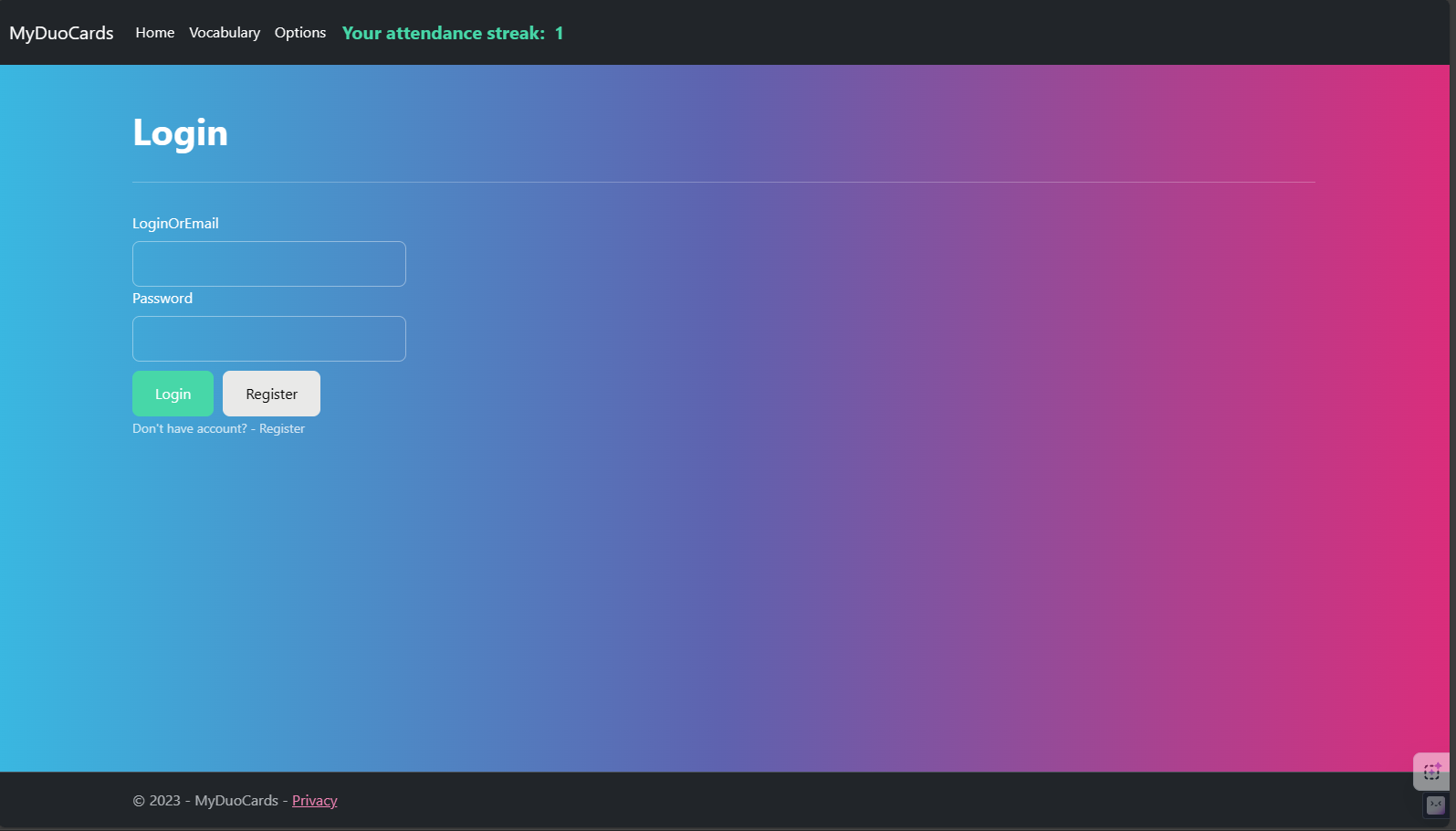


Рисунок - 8 Страница авторизации



Рисунок - 9 Страница регистрации

* Cтраница “Home”:

После успешной авторизации вы попадете на домашнюю страницу, где отображаются ваши карточки, логин, кнопка выхода из аккаунта и форма для фильтрации карточек. Вы также найдете кнопки, позволяющие просмотреть все ваши карточки, если их количество не вмещается на одной странице.

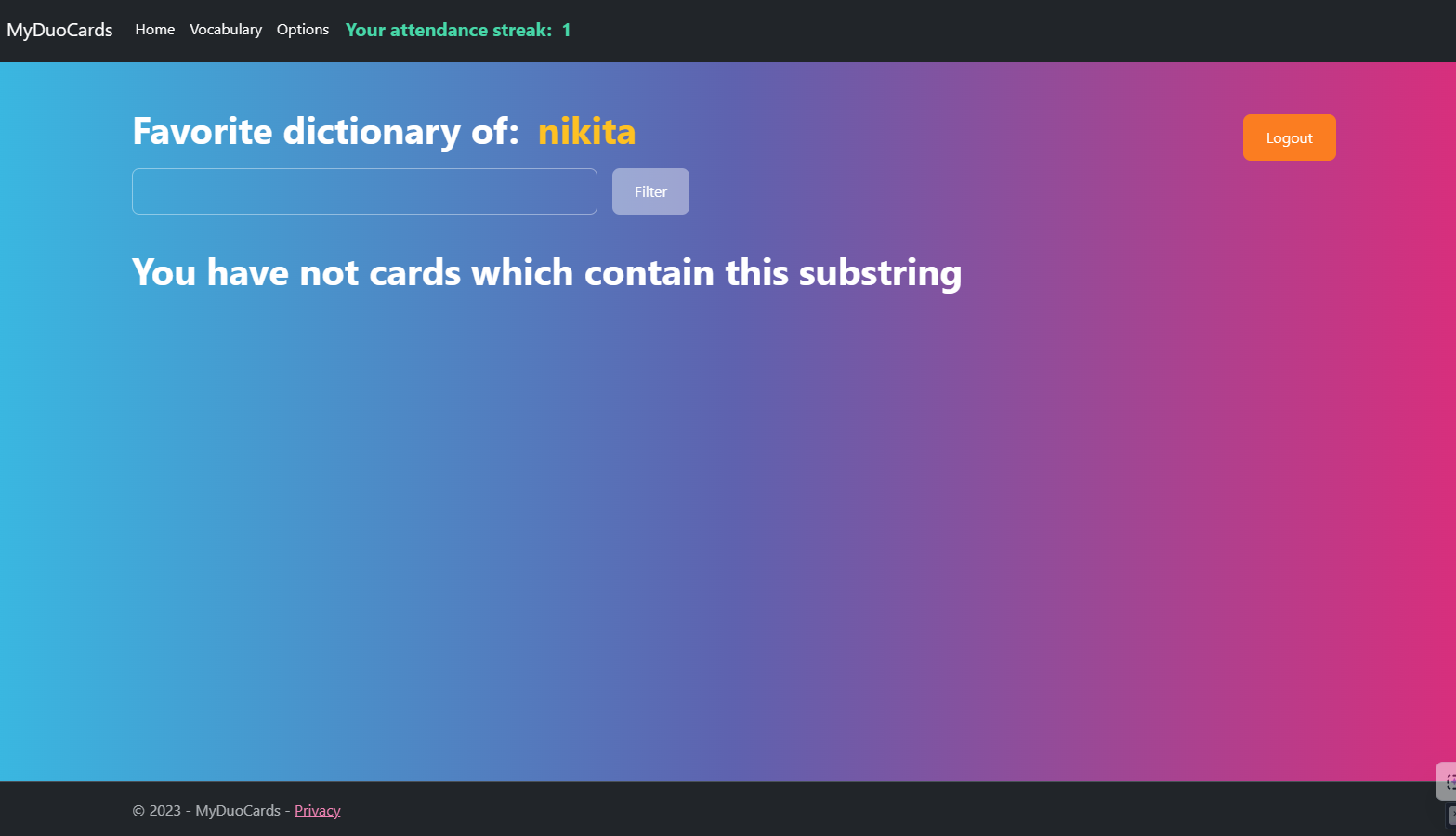


Рисунок - 10 Страница “Home”

* Страница "Vocabulary":

Эта страница содержит словарь всего приложения и подобна домашней странице, но без информации о конкретном пользователе или его карточках.

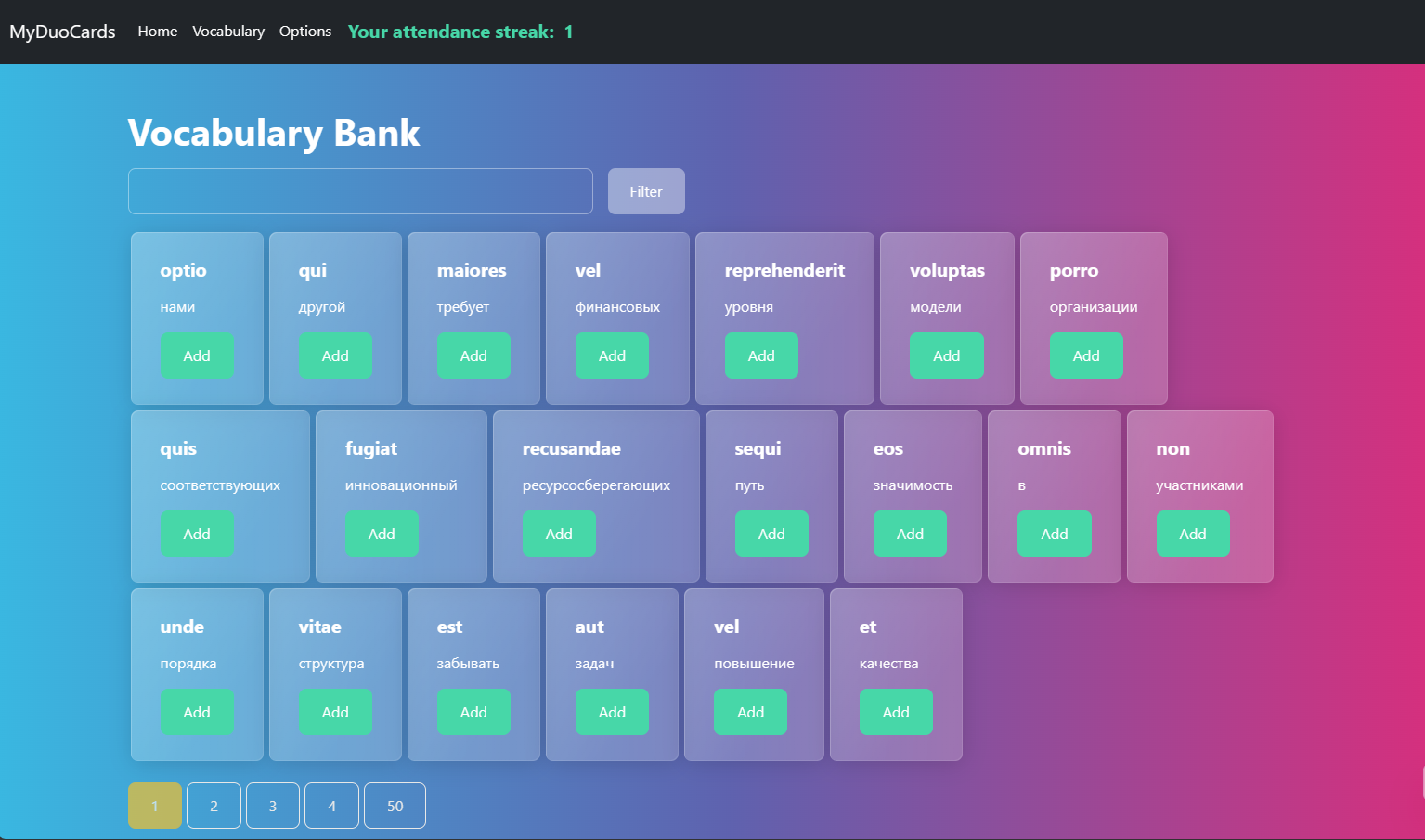


Рисунок - 11 Страница "Vocabulary"

* Страница “Options”. Здесь можно изменить различные данные о себе:

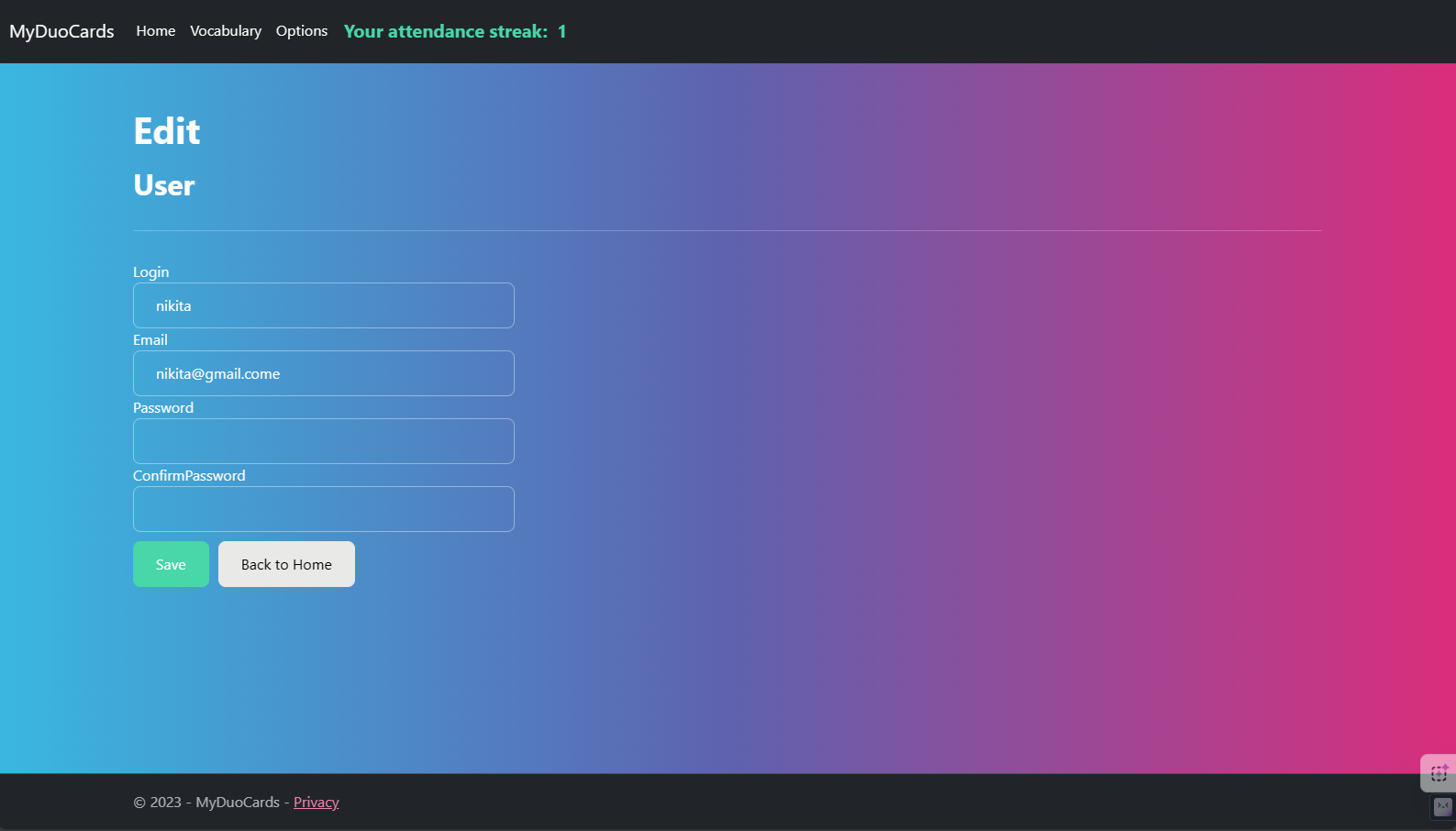


Рисунок - 12 Страница "Options"

Обратите внимание, что большая часть логики навигации располагается на частичной странице "layout", которая является основой для каждой из страниц и обладает различными переходами в зависимости от авторизации и роли пользователя.

**6.2 Руководство для администратора**

В приложение роль администратора поддерживает все возможности, присущие обычным пользователям, но при этом имеет дополнительные.

Информацию о функциональности, доступной пользователю, можно найти в пункте “6.1 Руководство обычного пользователя”. Следующие CRUD-страницы описаны на примере страниц одной сущности “EnWords”.

* На странице "Index" вы увидите полный список записей по определенной модели и сможете использовать поиск и пагинацию.

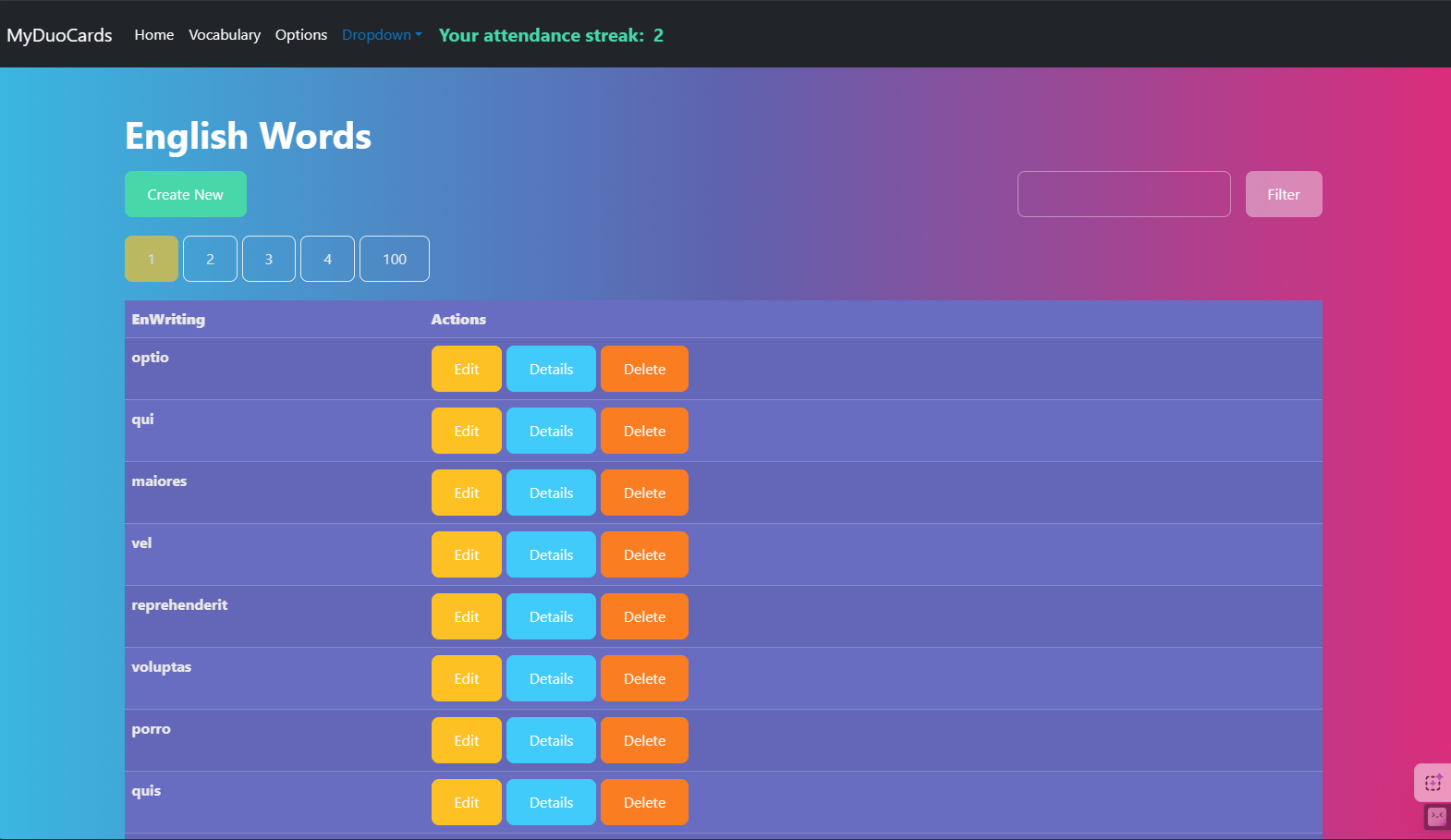


Рисунок - 13 Страница "Index"

* Страница "Details" позволяет просмотреть конкретную запись сущности и перейти к редактированию.

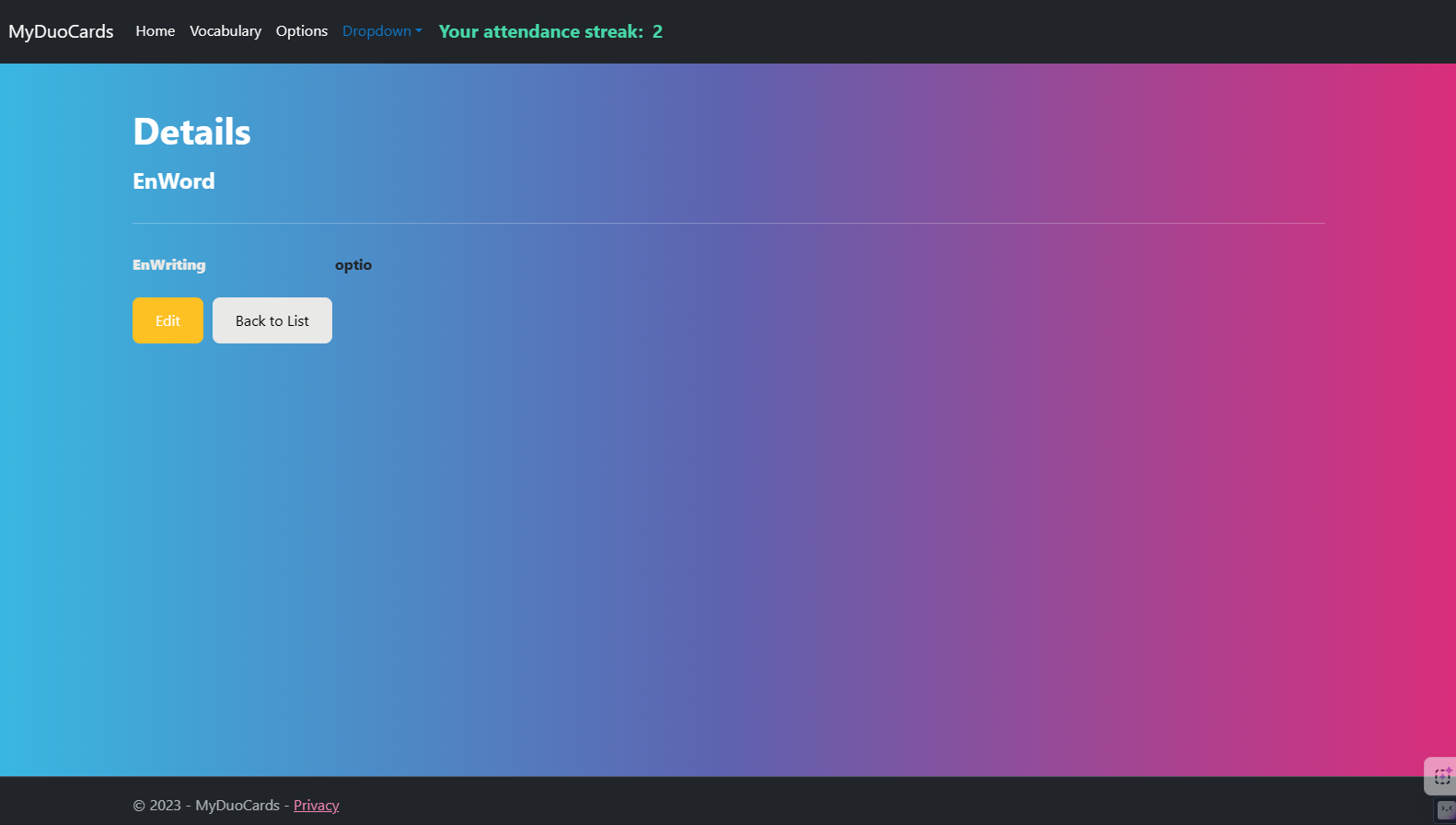


Рисунок - 14 Страница "Details"

* Затем страница "Edit" дает вам возможность редактировать конкретную запись.

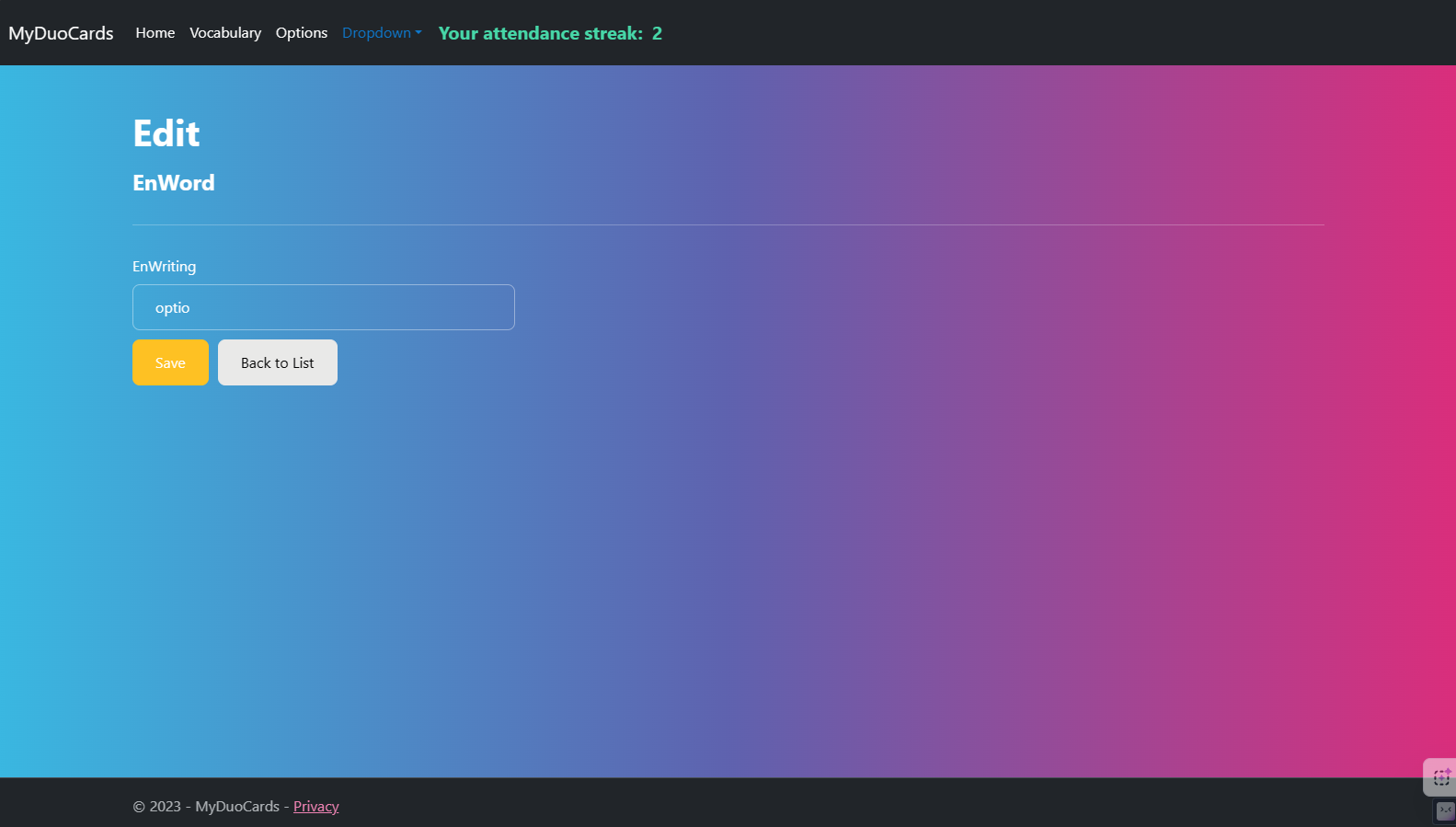


Рисунок - 15 Страница "Edit"

* Страница "Delete" предназначена для удаления конкретной записи.

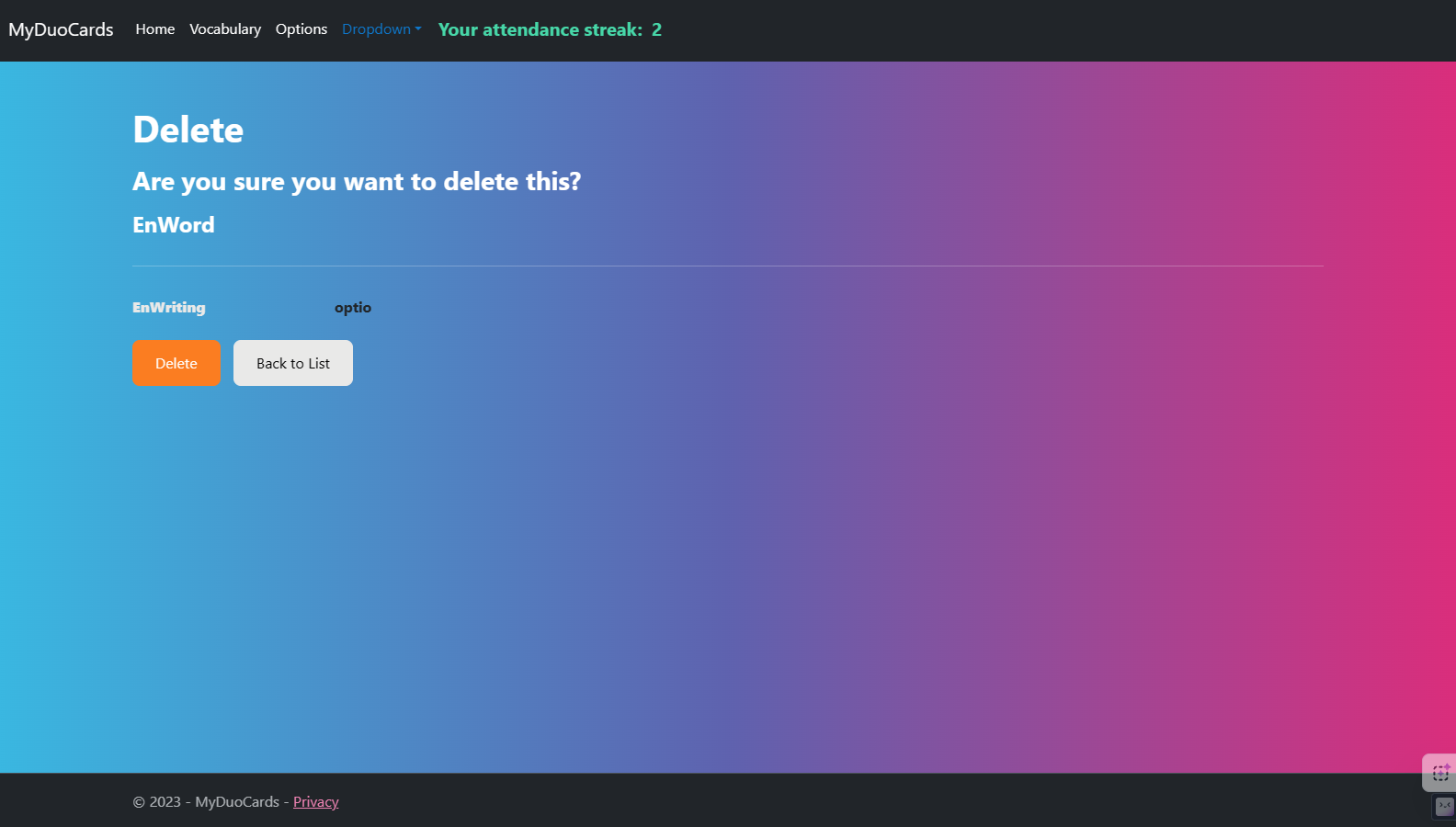


Рисунок - 16 Страница "Delete"

* Наконец, страница "Create" обеспечивает логику добавления новых записей к сущности в базе данных

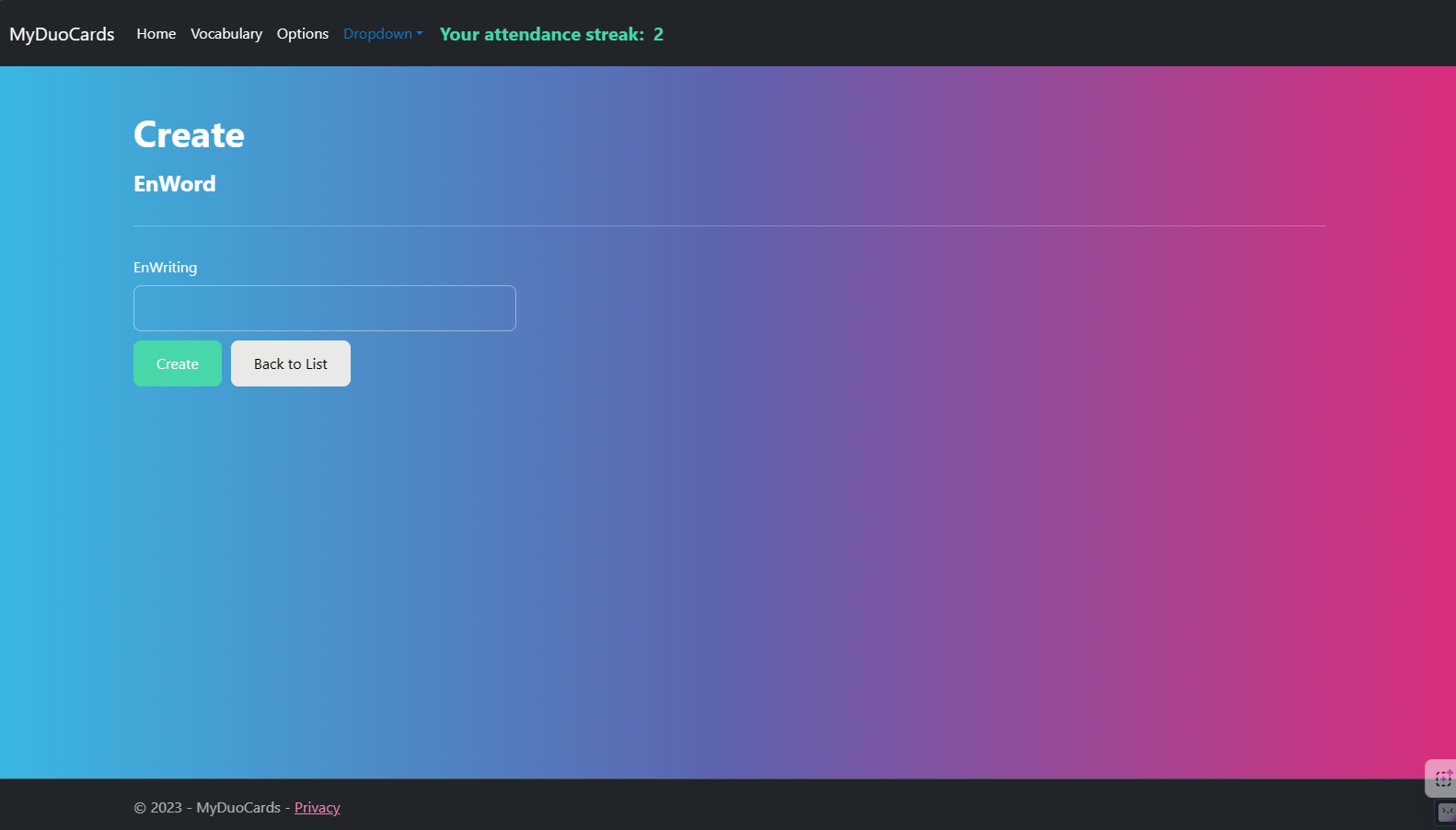


Рисунок - 17 Страница "Create"

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было разработано веб-приложения для изучения иностранных языков. Приложение реализовано средствами фреймворка ASP.NET Core, базы данных SQLite.

В результате выполнения данного проекта были решены задачи и соблюдены все требования, поставленные в начале работы над приложением.

На основе анализа результатов можно сделать вывод о том, что разработанное приложение и все входящие в его состав объекты работают корректно.

В будущем данное ПО можно дорабатывать до идеала, дополняя его более широким функционалом, выводить его на хостинг. В общем и целом, проводить всяческие улучшения и расширять проект, поскольку его актуальность практически безгранична в современном мире цифровых технологий и интернета.

Также разработчик получил обилие опыта работы с веб-технологиями, приобрел первые навыки с взаимодействием и использованием базы данных, а также ознакомился с самыми базовыми функциями контроля версий Git.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Сайт о программировании “Methanit” [Электронный ресурс] Руководство по ASP.NET Core MVC / Copyright © metanit.com, 2023. Все права защищены. / URL=”<https://metanit.com/sharp/aspnetmvc/>”
2. Официальная документация “Microsoft” [Электронный ресурс] Центр документации Entity Framework / © Microsoft 2023 / URL=”<https://learn.microsoft.com/ru-ru/ef/>”
3. Официальная документация “Microsoft” [Электронный ресурс] Документация по ASP.NET / © Microsoft 2023 / URL=”[Документация по ASP.NET | Microsoft Learn](https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-8.0)”