

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Зав. кафедрой «Математика

и информатика» ЧОУВО

«МИДиС», к.т.н., доцент

_____ / Л.Ю. Овсяницкая /

« _____ » _____ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой «ИТЭ»,

д.т.н., с.н.с.

_____ / Б.М. Суховилов /

« _____ » _____ 2020 г.

ИНТЕРАКТИВНЫЙ САЙТ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 09.03.03.2020.326.ВКР

Руководитель, к.т.н., доцент

_____ / И.А. Прохорова /

« _____ » _____ 2020 г.

Автор

студент группы ЭУ – 462

_____ / В. И. Селютин /

« _____ » _____ 2020 г.

Нормоконтролер, доцент

_____ / Е.А. Конова /

« _____ » _____ 2020 г.

Челябинск 2020

АННОТАЦИЯ

Селютин В. И. Интерактивный сайт дистанционного обучения – Челябинск: ЮУрГУ, ЭУ-462, 46с., 49 ил., 2 табл., библиогр. список –7 наим.

Выпускная квалификационная работа выполнена с целью разработки веб-приложения «Интерактивный сайт дистанционного обучения».

В ходе работы проведен анализ предметной области. Смоделирована и спроектирована структура сайта.

В результате работы создан интерактивный сайт для дистанционного обучения, позволяющий создавать собственные курсы или получать новые знания. Веб-приложение предоставляет познавательный контент, а также способствует в приобретении необходимых навыков онлайн на обучающей платформе.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1 Техничко-экономическая характеристика предметной области	10
1.2 Анализ аналогичных проектов	11
1.3 Анализ существующих инструментов разработки	13
1.4 Обоснование выбора средств разработки программного продукта.....	15
Выводы по разделу один	21
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО РЕШЕНИЯ	22
2.1 Постановка задачи	22
2.2 Краткое моделирование предметной области	22
Вывод по разделу два	29
3 РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ	30
3.1 Описание программной реализации	30
3.1.1 Создание проекта	30
3.1.2 Создание контекста Entity Framework.....	32
3.1.3 Создание базы данных	34
3.1.4 Разработка контроллеров.....	38
3.1.5 Подключение пакета Identity и настройка страниц авторизации.....	41
3.2 Руководство пользователя	47
3.2.1 Главная страница сайта.....	47
3.2.2 Страница подбора курсов.....	49
3.2.3 Режим обучающегося.....	50
3.2.4 Режим преподавателя.....	51
Выводы по разделу три.....	52
4 ЭКОНОМИЧЕСКИЙ АНАЛИЗ ПРОЕКТА.....	53
Вывод по разделу четыре	54
ЗАКЛЮЧЕНИЕ	55
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	56

ВВЕДЕНИЕ

Скорость развития современного мира и общества постоянно растет, поэтому человеку необходимо получать все больше новых знаний. Для многих людей намного удобней получать нужную информацию или приобретать необходимые навыки онлайн на обучающих платформах.

Разработанный интерактивный сайт для обучения позволит предоставлять познавательный контент и даст пользователям возможность принимать участие в функционировании веб-приложения. В ходе интерактивного обучения пользователь может получить не только теоретические знания, но и практические навыки. В гонке за знаниями важен не только объем приобретенной информации, но и скорость её получения, поэтому актуальность разработки очевидна, ведь с каждым днем растет потребность в обучающих веб-приложениях.

Постановка задачи – разработать современный сайт интерактивного обучения, с возможностью создания собственных курсов.

Объектом исследования является интернет ресурсы для интерактивного обучения.

Предмет исследования: процесс создания интерактивного сайта для обучения.

Целью выпускной квалификационной работы является проектирование и разработка интерактивного сайта для обучения.

Цель работы определила необходимость решения следующих задач:

- провести анализ предметной области;
- спроектировать информационную модель объекта;
- разработать сайт для интерактивного обучения.

Гипотеза исследования: после проектирования и разработки сайта мы получим эффективный инструмент для обучения.

Практическая значимость исследования заключается в том, что данным сайтом сможет пользоваться любой человек, который хочет получать знания или

создавать собственные курсы. Также возможно дальнейшее продвижение интерактивного сайта для обучения с целью получения экономической выгоды.

При выполнении выпускной квалификационной работы использовались теоретические и эмпирические методы исследования, такие как анализ, обобщение, конкретизация, изучение литературы, документов и результатов деятельности, анализ документов, обследование, изучение и обобщение опыта.

При подготовке выпускной квалификационной работы использовались учебники, монографии отечественных и зарубежных авторов; материалы периодических изданий.

Выпускная квалификационная работа состоит из трех глав.

В первой главе рассмотрена предметная область, проведен анализ аналогичных проектов, обоснован выбор средств разработки программного продукта.

Во второй главе смоделирована и спроектирована структура сайта.

В третьей главе описан процесс разработки сайта.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Техничко-экономическая характеристика предметной области

Интерактивный – означает способность взаимодействовать или находится в режиме беседы, диалога с кем-либо (человеком) или чем-либо (например, компьютером). Следовательно, интерактивное обучение – это, прежде всего, диалоговое обучение, в ходе которого осуществляется взаимодействие преподавателя и обучающегося.

Интерактивный сайт – это динамический, постоянно изменяющийся по содержанию ресурс, изменения в котором производят непосредственно сами пользователи сайта. Создание интерактивных сайтов предполагает разработку интерактивных элементов, при помощи которых происходит взаимодействие с пользователями (форумы, гостевые страницы и т.д.). Такие сайты отлично подойдут для предоставления обучающего контента, так как дают пользователям или ученикам полную возможность принимать активное участие в функционировании веб-приложения.

Суть интерактивного обучения через обучающую онлайн-платформу состоит в том, что учебный процесс организован таким образом, что практически все обучающиеся оказываются вовлеченными в процесс познания, они имеют возможность понимать и рефлексировать по поводу того, что они знают и думают. Совместная деятельность обучающихся в процессе познания, освоения учебного материала означает, что каждый вносит свой особый индивидуальный вклад, идет обмен знаниями, идеями, способами деятельности. Причем, происходит это в атмосфере доброжелательности и взаимной поддержки, что позволяет не только получать новые знания, но и развивает саму познавательную деятельность, переводит ее на более высокие формы кооперации и сотрудничества.

Преподаватель не дает готовых знаний, он побуждает обучаемых к самостоятельному поиску. По сравнению с традиционными формами ведения занятий, в интерактивном обучении через обучающую онлайн-платформу

меняется взаимодействие преподавателя и обучаемого: активность педагога уступает место активности обучаемых, а задачей педагога становится создание условий для их инициативы.

Создать интерактивный сайт для обучения – значит создать сайт, умеющий «общаться» с пользователями, получающими новые знания. На простом статическом или информационном сайте доступен лишь просмотр контента. В то же время, интерактивный сайт позволит поддерживать активную связь между администрацией и посетителями или учителем и учеником в режиме реального времени без каких-либо вспомогательных средств – для этого есть системы онлайн-консультаций, онлайн-чатов (системы мгновенного обмена сообщениями прямо на сайте) и другие подобные инструменты.

1.2 Анализ аналогичных проектов

Существует множество образовательных сайтов. Их большая часть имеет большой спектр направлений, базу с обучающим материалом и интуитивно понятный интерфейс. В нашем случае рассмотрены сайты, имеющие платформы для обратной связи с преподавателем, либо для интерактивного обучения.

ALISON – это платформа с огромным количеством курсов разного направления: от управления бизнесом до изучения языков. Сайт предоставляет возможность сдать онлайн-экзамен и получить бесплатный виртуальный сертификат.

Сайт предоставляет курсы и тренинги по таким направлениям как:

- информационные технологии;
- лингвистика;
- здоровье;
- гуманитарные науки;
- бизнес;
- естественные науки;
- маркетинг;

- Lifestyle (включающие в себя курсы по фотографии, музыке и т.д.).

Из преимуществ сайта можно выделить удобный и интуитивно понятный интерфейс, большой выбор направлений.

Одним из главных недостатков для русских пользователей является отсутствие поддержки русского языка на сайте и в курсах. Доступна поддержка только английского, испанского, французского, итальянского и бразильский вариант португальского языка.

edX – бесплатные лекции от преподавателей ведущих учебных заведений (Гарвард, Беркли и другие). Сайт создан специально для интерактивного обучения через Интернет.

Основные направления курсов на сайте:

- информационные технологии;
- лингвистика;
- статистика;
- бизнес и менеджмент;
- инженерия;
- гуманитарные науки.

К преимуществам сайта можно отнести удобный интерфейс, множество направлений и возможность интерактивного обучения. Сайт сотрудничает с множеством зарубежных университетов.

Главными недостатками сайта являются отсутствие русского языка и возможности создания своего собственного курса.

«**Универсариум**» – это российская открытая система электронного образования. Сайт сотрудничает с лучшими вузами России и имеет множество курсов в разных направлениях.

Основные направления на сайте:

- гуманитарные науки;
- естественные науки;
- педагогика;

- саморазвитие;
- передовые технологии (космические и нано-технологии).

К плюсам можно отнести современный дизайн, множество направлений и курсов. Присутствует рейтинг обучающихся.

Из минусов можно выделить отсутствие возможности создавать свои собственные курсы и высокую стоимость обучения.

Сравнительный анализ аналогичных проектов представлен в таблице 1.

Таблица 1 – Сравнительный анализ аналогичных проектов

Показатель	«ALISON»	«edX»	«Универсариум»
Простота интерфейса	Сложный интерфейс	Сложный интерфейс	Сложный интерфейс
Поддержка русского языка	Нет	Нет	Да
Возможность создания собственных функций	Да	Нет	Нет
Стоимость прохождения курса на веб-ресурсе	Бесплатно	Бесплатно	От 1000 за один курс

По результатам анализа сделан вывод о том, что большая часть аналогичных проектов имеют сложный интерфейс и широкий функционал, не все сайты имеют возможность создания собственных курсов. Таким образом, разрабатываемый сайт должен иметь простой интерфейс, с широким функционалом и возможностью создания собственных курсов.

1.3 Анализ существующих инструментов разработки

Для поставленной задачи требуется провести анализ существующих инструментов разработки с описанием основных их характеристик. Учитывая важные критерии, можно подобрать оптимальные инструменты для решения конкретно поставленной задачи за небольшой промежуток времени и с эффективным результатом.

Для этого проанализируем следующие оптимальные продукты, подходящие для разработки программных решений: «Microsoft Visual Studio. 2019» и Zend Studio.

1. «Microsoft Visual Studio. 2019».

«Microsoft Visual Studio» – это продукт компании «Microsoft», включающий интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых «Windows», Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight [9].

«Visual Studio» включает в себя редактор исходного кода с возможностью контролируемого процесса улучшения кода, например, предлагает переименовать функцию или добавить параметр. Встроенный отладчик может работать как отладчик исходного уровня, так и как отладчик машинного уровня. Остальные инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор.

«Visual Studio» позволяет создавать и подключать различные сторонние дополнения для расширения функциональности, включая добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент «Team Explorer»), разработка мобильных приложений и игр.

2. «Zend Studio».

«Zend Studio» – проприетарная интегрированная среда разработки приложений (IDE) на языке программирования PHP, разработанная «Zend Technologies». Основана на «PHP Development Tools» для «Eclipse».

«Zend Studio» тесно интегрирована с «Zend Server», что предоставляет удобную систему отладки PHP-приложений. Также «Zend Studio» интегрирована с «Zend Framework» и содержит инструменты для упрощения работы с этим фреймворком.

1.4 Обоснование выбора средств разработки программного продукта

Для достижения поставленной цели в качестве выбора среды разработки выбран «Microsoft Visual Studio», предназначенный для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб, а в качестве технологии соответственно ASP.NET. Язык реализации проекта – C#. В качестве СУБД мы используем Microsoft SQL Server 2016. Также рассмотрим нужные технологии и библиотеки в реализации проекта – AJAX и библиотеку jQuery.

1. «Microsoft Visual Studio. 2019».

Главными достоинствами Visual Studio являются создание, как собственного кода, так и управляемого кода, Visual Studio включает в себя редактор кода, поддерживающий IntelliSense, а также рефакторинг кода.

Интегрированный отладчик работает как отладчик уровня источника и отладчик уровня машины. Другие встроенные инструменты включают конструктор форм для создания приложений с графическим интерфейсом, веб-дизайнер, дизайнер классов и конструктор схем базы данных. Он принимает плагины, которые расширяют функциональные возможности практически на каждом уровне, включая добавление поддержки для систем контроля версий (таких как Subversion) и добавление новых наборов инструментов, таких как редакторы и визуальные дизайнеры для языков, специфичных для предметной

области, или наборов инструментов для других аспектов жизненного цикла разработки программного обеспечения. Например, клиент Team Foundation Server: Team Explorer.

Visual Studio поддерживает различные языки программирования и позволяет редактору кода и отладчику поддерживать (в разной степени) практически любой язык программирования при условии, что существует специальная языковая служба. Встроенные языки включают C, C++ и C++ / CLI (через Visual C++), VB.NET (через Visual Basic .NET), C# (через Visual C#) и F# (начиная с Visual Studio 2010). Он также поддерживает XML / XSLT, HTML / XHTML, JavaScript и CSS, что очень важно для создания сайта.

2. Язык реализации C#.

В качестве языка реализации выбран язык C# ввиду следующих его преимуществ:

- C# является объектно-ориентированным языком;
- язык C# разрабатывался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности;
- C# является наследником языков C/C++, эти языки имеют общий синтаксис, что облегчает переход от C++ к C#;
- мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
- простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как адресная арифметика, адресация, разыменованное, и указатели.

3. Microsoft SQL Server 2016.

Microsoft SQL Server в качестве языка запросов использует версию SQL, получившую название Transact-SQL (сокращённо T-SQL), являющуюся реализацией SQL-92 (стандарт ISO для SQL) с множественными расширениями. TSQL позволяет использовать дополнительный синтаксис для хранимых

процедур и обеспечивает поддержку транзакций (взаимодействие базы данных с управляющим приложением). Microsoft SQL Server и Sybase ASE для взаимодействия с сетью используют протокол уровня приложения под названием Tabular Data Stream (TDS, протокол передачи табличных данных).

Microsoft SQL Server также поддерживает Open Database Connectivity (ODBC) – интерфейс взаимодействия приложений с СУБД. Версия SQL Server 2016 обеспечивает возможность подключения пользователей через веб-сервисы, использующие протокол SOAP. Это позволяет клиентским программам, не предназначенным для Windows, кроссплатформенно соединяться с SQL Server.

В SQL Server 2016 встроена поддержка .NET Framework. Благодаря этому, хранимые процедуры БД могут быть написаны на любом языке платформы .NET, используя полный набор библиотек, доступных для .NET Framework, включая Common Type System (система обращения с типами данных в Microsoft .NET 2.0 Framework). Однако, в отличие от других процессов, .NET Framework, будучи базисной системой для SQL Server 2016, выделяет дополнительную память и выстраивает средства управления SQL Server вместо того, чтобы использовать встроенные средства Windows. Это повышает производительность в сравнении с общими алгоритмами Windows, так как алгоритмы распределения ресурсов специально настроены для использования в структурах SQL Server ASP.NET.

4. Технология ASP.NET.

ASP.NET – технология веб-программирования от компании Microsoft. ASP.NET - развитие более старой технологии Microsoft ASP, но имеет ряд отличительных особенностей. Код для ASP.NET можно писать, используя практически любые языки программирования, в том числе, и входящие в комплект .NET Framework (C#, Visual Basic.NET и JScript .NET). ASP.NET имеет преимущество в скорости по сравнению со скриптовыми технологиями, так как при первом обращении код компилируется и помещается в специальный кэш, и впоследствии только исполняется. ASP.NET предлагает новую модель

программирования и инфраструктуру, которые позволяют разрабатывать защищенные и масштабируемые решения[4].

Для работы с ASP.NET необходимо установить соответствующее программное обеспечение – Microsoft Visual Studio. В этом редакторе необходимо создать файл – страницу веб-ресурса. Основная часть такого файла представляет собой обычную структуру HTML-документа. В самом начале этого документа размещена строка, содержащая директивы ASP.NET. Также в начале файла находится информация об используемом языке программирования в модуле, связанном со страницей (в нашем случае это C#), об автоматическом связывании событий страницы с функциями – обработчиками этих событий, а также имя модуля, который содержит программный код, привязанный к странице. Наличие расширения .aspx у файла говорит о том, что он содержит некие директивы ASP.NET и должен быть исполнен платформой .NET Framework.

В отличие от языка PHP, который поддерживает различные операционные системы и СУБД, ASP.NET наиболее полно взаимодействует с продуктами компании Microsoft – операционными системами семейства Windows и СУБД Microsoft SQL Server или Microsoft SQL Server Express [5].

Выбрав самостоятельную разработку на ASP.NET, это окупится возможностями масштабирования, большой гибкостью, повышенной безопасностью и меньшей стоимостью разработки и поддержки в сравнении с аналогами.

Используя свой движок, разработанный на ASP.NET и правильный подход к построению архитектуры решения возможно дальнейшее развитие проекта и расширение его функционала без больших усилий.

При разработке на ASP.NET можно на этапе отладки выделять ресурсоёмкие механизмы и принимать меры для оптимального использования ресурсов сервера. Поэтому можно использовать менее затратный хостинг.

5. AJAX.

AJAX (Asynchronous JavaScript and XML) – подход к построению пользовательских интерфейсов веб-приложений, при котором веб-страница, не перезагружаясь, сама догружает нужные пользователю данные. AJAX – один из компонентов DHTML. Это посредник между браузером (загруженной в него страницей) и сервером, способный передавать данные серверу и принимать их от него [6].

AJAX – это идея, которая базируется на двух основных принципах:

- использование DHTML для динамического изменения содержания страницы;
- использование XMLHttpRequest для быстрого обращения к серверу.

Использование этих двух подходов позволяет создавать намного более удобные веб-интерфейсы пользователя на тех страницах сайтов, где необходимо активное взаимодействие с пользователем, что прямо определяет цель создания нашего сайта.

6. Библиотека jQuery.

jQuery – это популярная JavaScript библиотека. Библиотека jQuery содержит функционал, полезный для максимально широкого круга задач.

Библиотека jQuery в первую очередь обеспечивает непротиворечивую работу программного кода во всех типах браузеров, решая такие сложные проблемы JavaScript, как ожидание загрузки страницы перед тем, как выполнять какие-либо операции [3]. На тот случай, если в библиотеке обнаружится недостаток функциональности, разработчики предусмотрели простой, но весьма действенный способ её расширения. Многие начинающие программисты jQuery обнаруживают эту гибкость на практике, расширяя возможности jQuery в первый же день. Чтобы привнести динамическую функциональность на любую интернет-страницу, приходится следовать одному и тому же шаблону: сначала отбирается элемент или группа элементов, а затем над ними выполняются некоторые действия, например, скрывать или показывать интересующие нас элементы, добавлять к ним класс CSS, создавать анимационные эффекты или изменять

атрибуты. С обычным JavaScript для решения каждой из задач потребуется десятки строк программного кода. Создатель jQuery разработал свою библиотеку именно для того, чтобы сделать наиболее общие задачи тривиальными. Например, чтобы создать таблицу с разным цветом фона для четных и нечетных строк, дизайнеру потребуется написать до 10 строк кода на языке JavaScript, а вот с использованием jQuery этот эффект достигается с использованием не более чем одной строки.

Выводы по разделу один

В первой главе выпускной квалификационной работы проанализирована технико-экономическая характеристика предметной области, проведен теоретический анализ предметной области, проведен анализ существующих программных продуктов с выявлением их достоинств и недостатков. Осуществлен анализ существующих инструментов разработки, а также обоснован выбор средств разработки и технологий проектирования.

Наше исследование показало необходимость в разработке собственного программного продукта, учитывающее достоинства и недостатки аналогичных продуктов в рассматриваемой предметной области.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО РЕШЕНИЯ

2.1 Постановка задачи

В результате разработки должен быть создан сайт интерактивного обучения с возможностью создания и ведения собственных курсов, обработки и сохранения результатов прогресса пользователя.

Сайт интерактивного обучения должен обеспечивать выполнение перечисленных ниже функций и задач:

- создание и редактирование новых пользовательских курсов;
- хранение пользовательских курсов;
- поиск курсов по выбранным пользователем критериям;
- возможность выполнения и редактирования практических работ;
- хранение результатов выполненных практических работ;
- возможность обратной связи пользователей;
- идентификация зарегистрированных пользователей;
- создание личного кабинета для каждого пользователя;
- хранение информации о пользователях;
- возможность создание закрытого курса, для ограничения прав

доступа.

Разрабатываемый сайт предназначен для людей, которым требуется интерактивная онлайн платформа для обучения.

2.2 Краткое моделирование предметной области

Любой сайт, не зависимо от его назначения, является сложной информационной системой, включающей в себя процессы, потоки данных и хранилище данных.

Существует достаточно большое количество методов моделирования процессов. Эти методы относятся к разным видам моделирования и позволяют сфокусировать внимание на различных аспектах. Они содержат как графические, так и текстовые средства, за счет которых можно наглядно представить основные

компоненты процесса и дать точные определения параметров и связей элементов нашей системы.

Мы используем а нотацию DFD, с использованием инструментария BPWin, которая позволит наглядно смоделировать информационную систему с точки зрения хранения, обработки и передачи данных.

DFD – общепринятое сокращение от англ. data flow diagrams – диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Диаграмма потоков данных (data flow diagram, DFD) – один из основных инструментов структурного анализа и проектирования информационных систем [11].

Непосредственно DFD нотация состоит из следующих элементов:

1. Процесс (англ. Process), т.е. функция или последовательность действий, которые нужно предпринять для обработки данных.
2. Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы.
3. Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе. Это и есть базы данных, таблицы или любой другой вариант организации и хранения данных.
4. Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая выходит из того или иного блока на диаграмме.

На рисунке 1 представлена общая диаграмма структуры деятельности сайта, которая отражает внешние сущности – «Преподаватели» и «Ученики», которые взаимодействуют с сайтом, основной процесс и потоки информации.

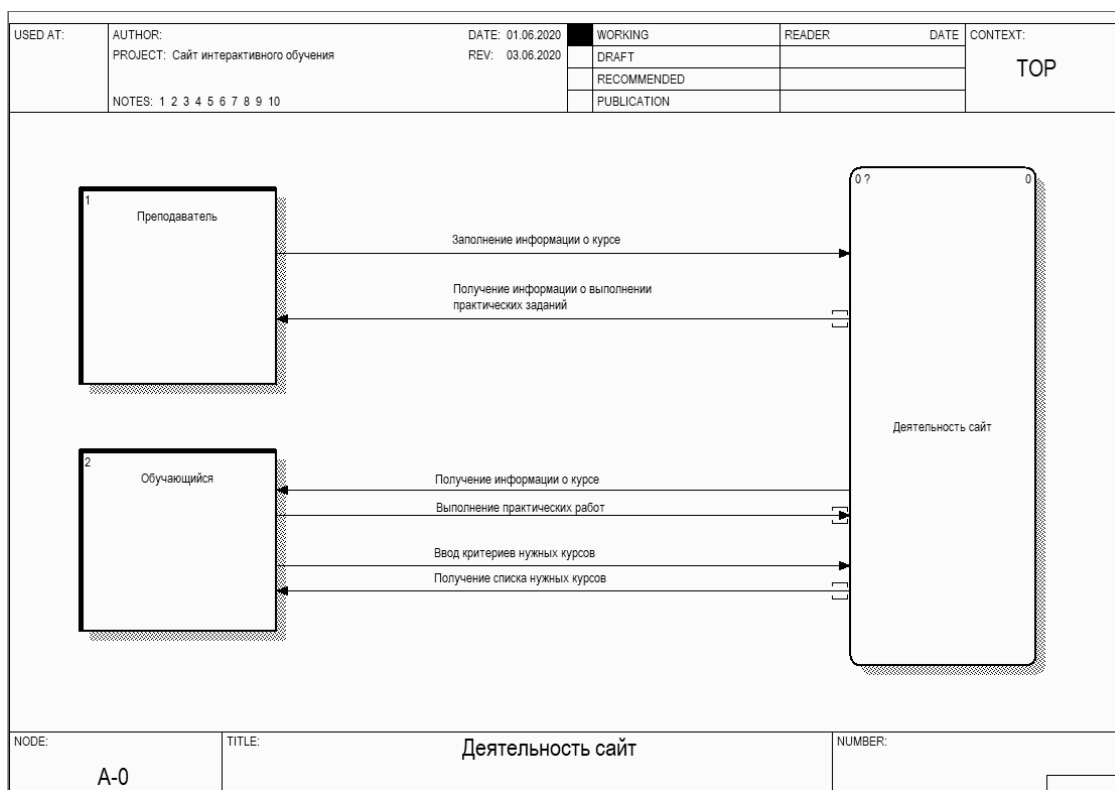


Рисунок 1 – Деятельность сайта

Любой процесс нужно детализировать. Декомпозиция диаграммы деятельность сайта представлена на рисунке 2, отражает все блоки процессов, проходящих на сайте.

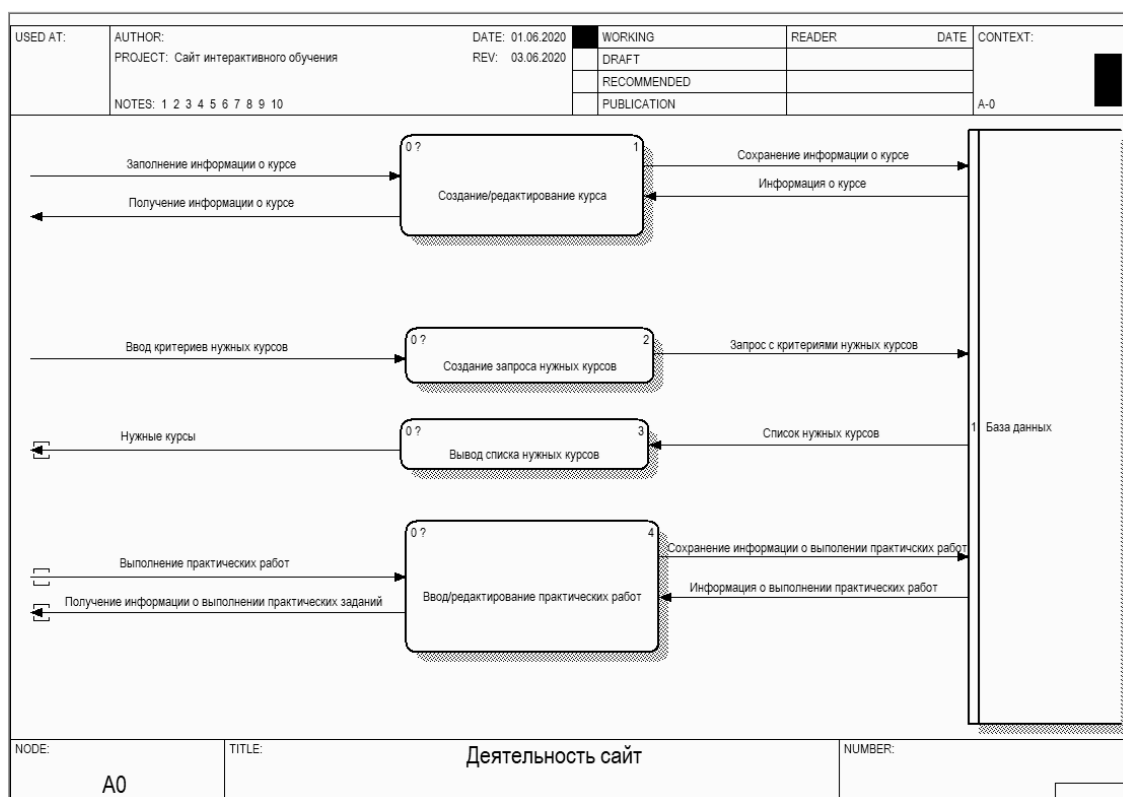


Рисунок 2 – Декомпозиция деятельности сайта

Смоделированы функции диаграммы функциональной декомпозиции.

На сайте необходима «Возможность создания/редактирования курса», «Ввода/ вывода результатов практических работ» и «Вывод списка курсов». В качестве объекта хранилища данных выступает наша «База данных».

В первой функции преподаватель создает или редактирует информацию о курсе и сохраняет ее в базу данных, а ученик получает «Информацию о курсе», эта информация выступает в качестве потоков данных.

Во второй функции ученики вводят информацию о нужных им курсах, на основе которой формируется запрос на сайте для вывода списка курсов. «Получение списка курсов» являются выходными потоками данных. Входным потоком данных является «Запрос с критериями нужных курсов».

В третьей функции ученики выполняют практические работы и сохраняют информацию в базе данных, затем преподаватель получает информацию о выполнении работ.

Подробно рассмотрим декомпозицию функции «Создание/редактирование курса» представленную на рисунке 3. Рассматриваемый процесс является неотъемлемой функцией работы сайта, преподаватель создает курс и может его редактировать.

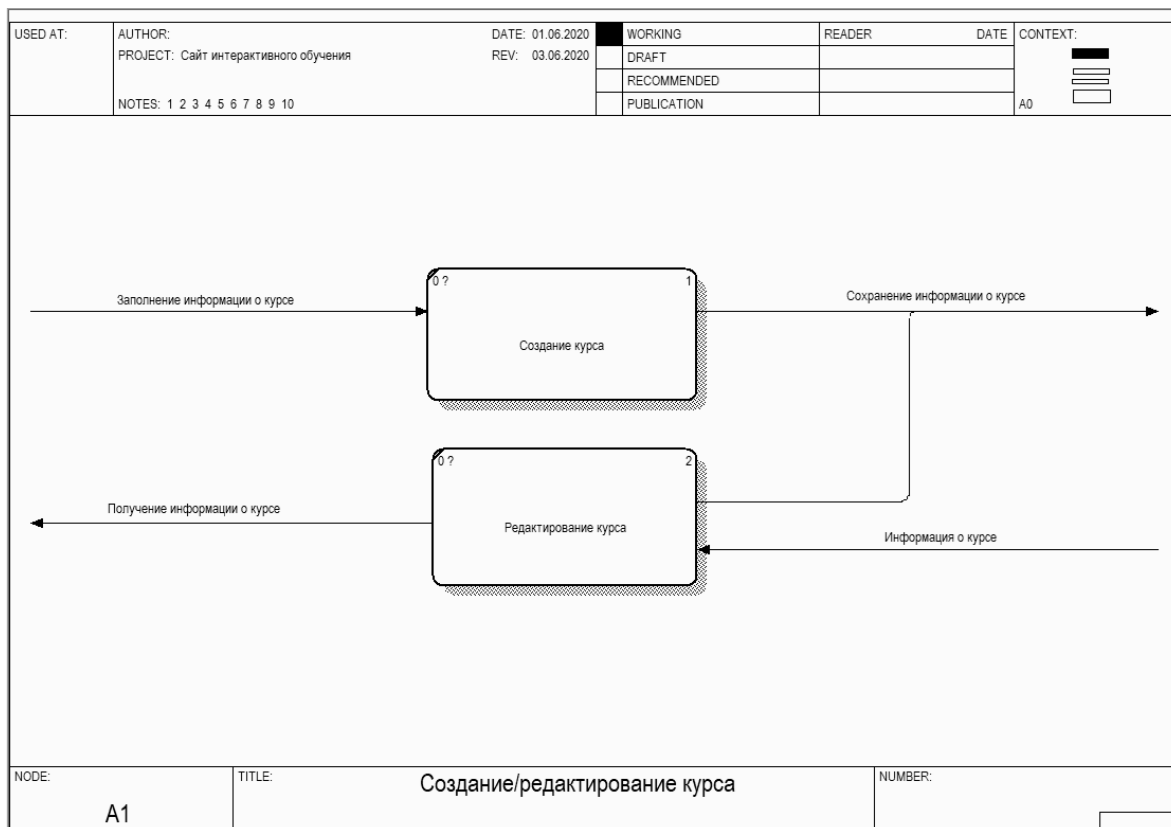


Рисунок 3 – Декомпозиция функции «Создание/редактирование курса»

Входным потоком данных функции «Создание курса» является «Заполнение информации о курсе», выходным – «Сохранение информации о курсе». У функции «Редактирование курса» входной поток информации «Информация о курсе», выходными потоками являются «Сохранение информации о курсе» и «Получение информации о курсе».

Декомпозиция «Создание запроса нужных курсов» представлена на рисунке 4. Рассматриваемая функция обрабатывает полученную информацию ученика о критериях нужных ему курсов.

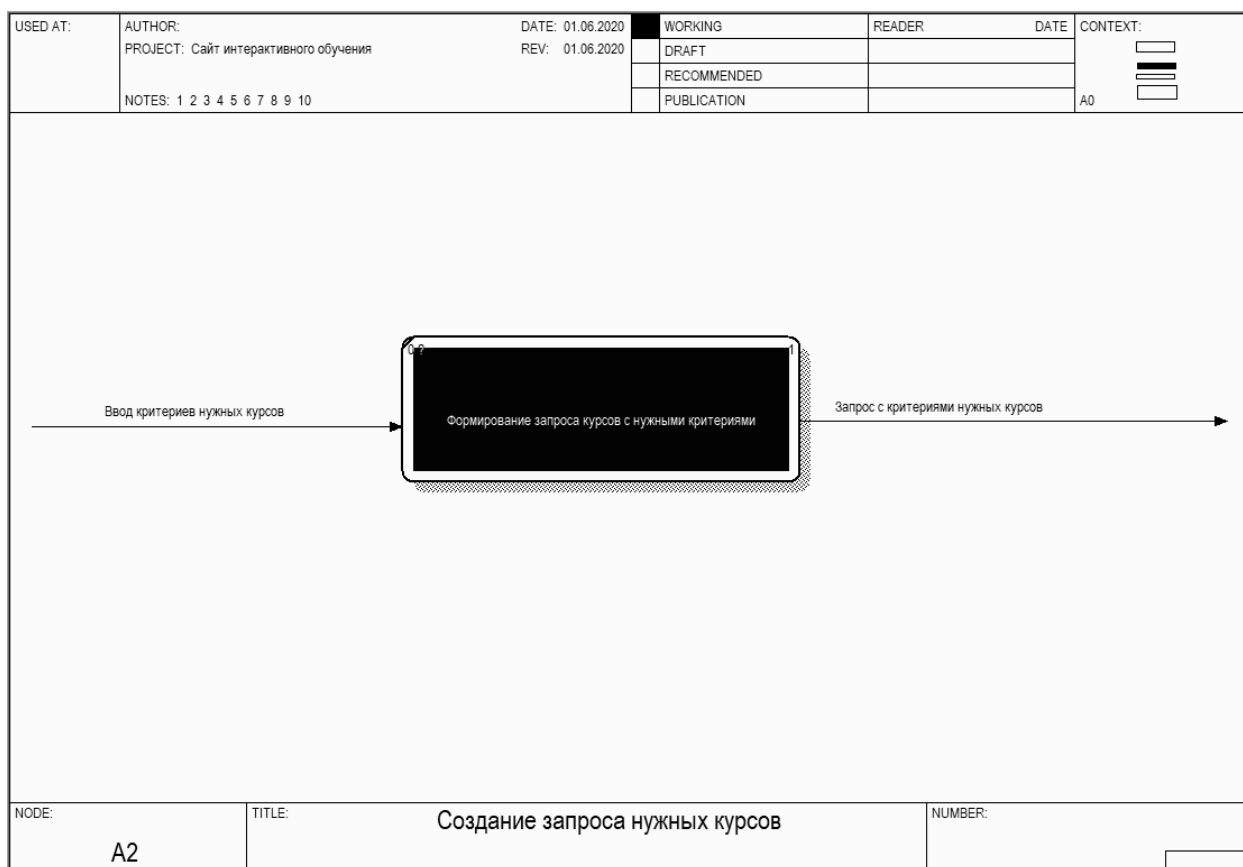


Рисунок 4 – Декомпозиция функции «Вывод списка курсов»

Декомпозиция «Вывод нужных курсов» представлена на рисунке 5. Рассматриваемая функция получает список содержащий информацию и курсах основываясь на критериях и формирует HTML код на основе запроса.

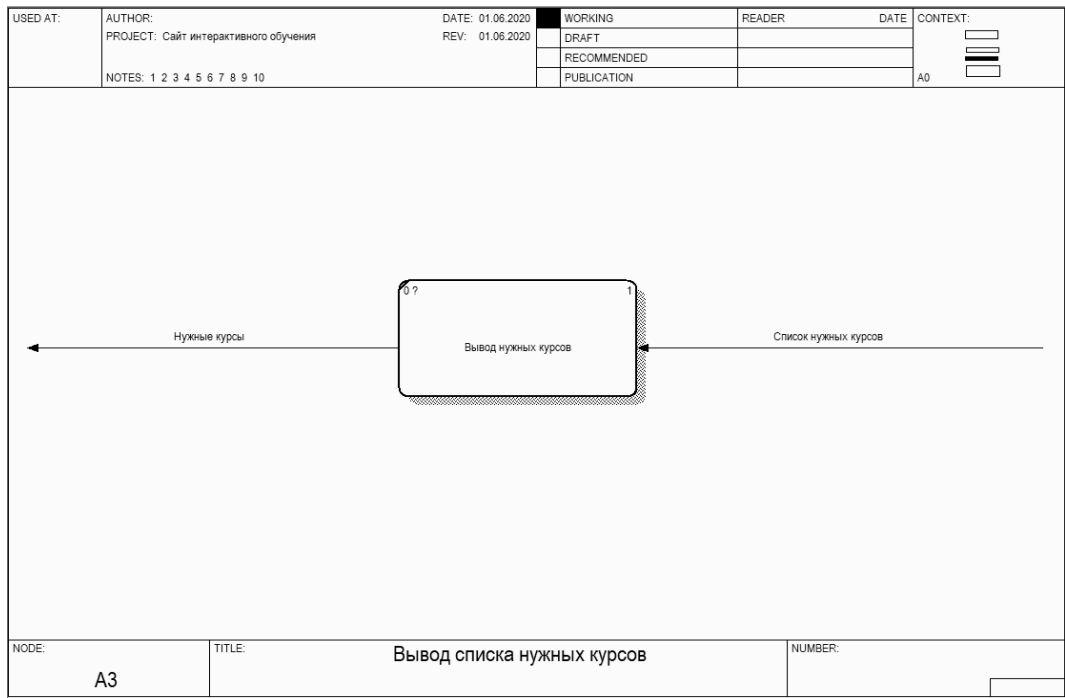


Рисунок 5 – Декомпозиция функции «Вывод списка курсов»

Декомпозиция «Ввод/ вывод результатов практических работ» представлена на рисунке 6. Рассматриваемая функция позволяет ученику сохранить результаты выполненных работ, а преподавателю получить информацию о них.

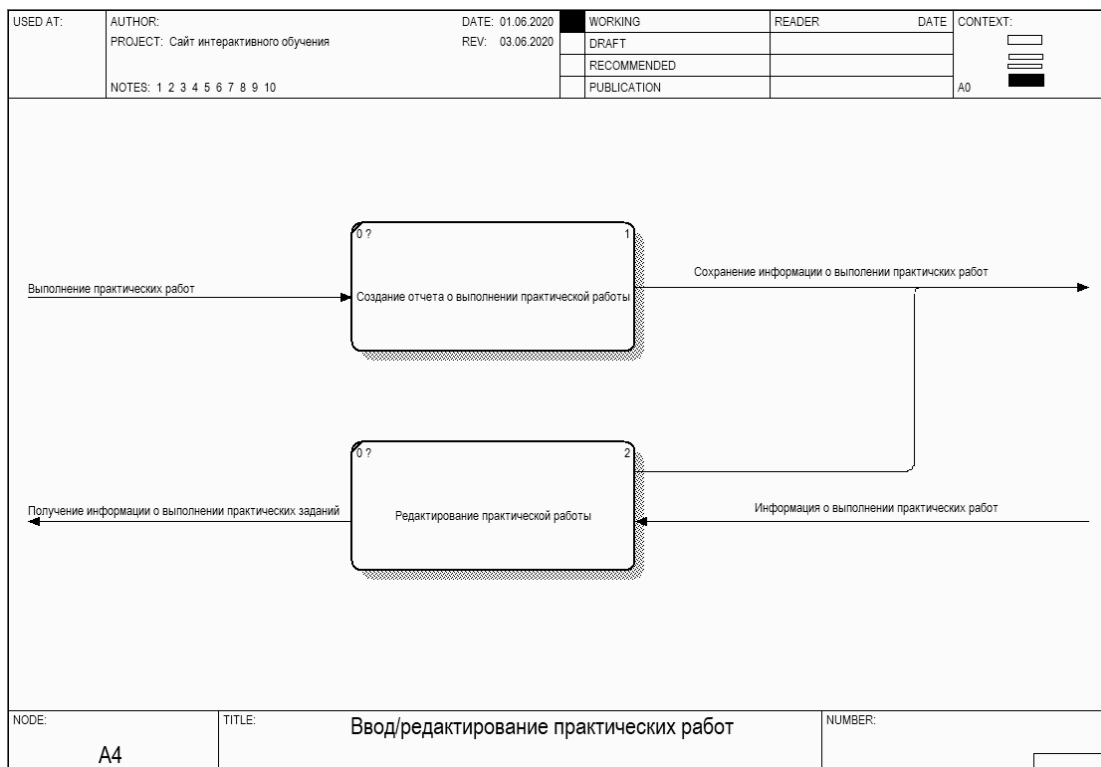


Рисунок 6 – Декомпозиция функции
«Ввод/ вывод результатов практических работ»

Вывод по разделу два

Во второй главе выпускной квалификационной работы определено основное назначение разрабатываемого сайта, а для программного решения смоделирована предметная область и разработана методология IDEF0, представляющая собой моделирование информационных систем с точки зрения хранения, обработки и передачи данных. На диаграммах изображены функции, потоки данных, хранилища данных и внешние сущности, которые отражают деятельность работы сайта интерактивного обучения.

3 РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ

3.1 Описание программной реализации

3.1.1 Создание проекта

Разработка веб-приложения осуществлена с помощью среды разработки «Microsoft Visual Studio 2019» и её технологии ASP.NET [4].

С помощью Visual Studio Installer мы установили последнюю версию Visual Studio с пакетами разработки ASP.NET, затем создали и настроили проект с поддержкой ASP.NET Core. При создании проекта выбран способ проверки подлинности учетных записей отдельных пользователей. Установка Visual Studio с пакетами разработки ASP.NET представлена на рисунке 7.

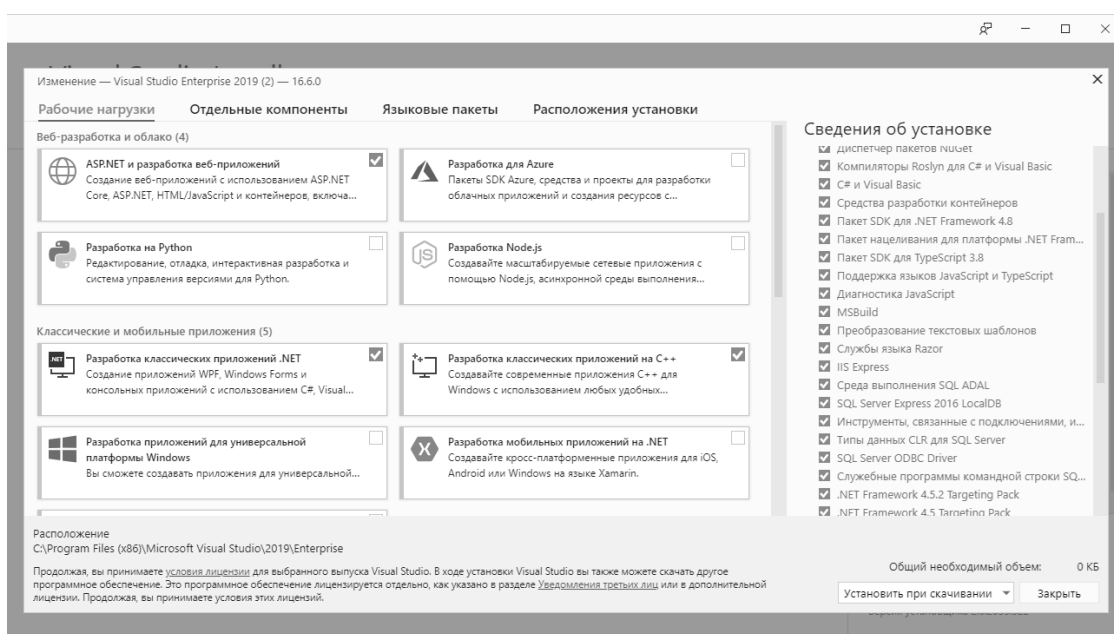


Рисунок 7 – Visual Studio Installer

Файлы и папки проекта приведены на рисунке 8. Основываясь на объектной модели приложения, мы создали модели, так же интерфейсы, для работы с моделями, и репозитории, реализующие интерфейсы.

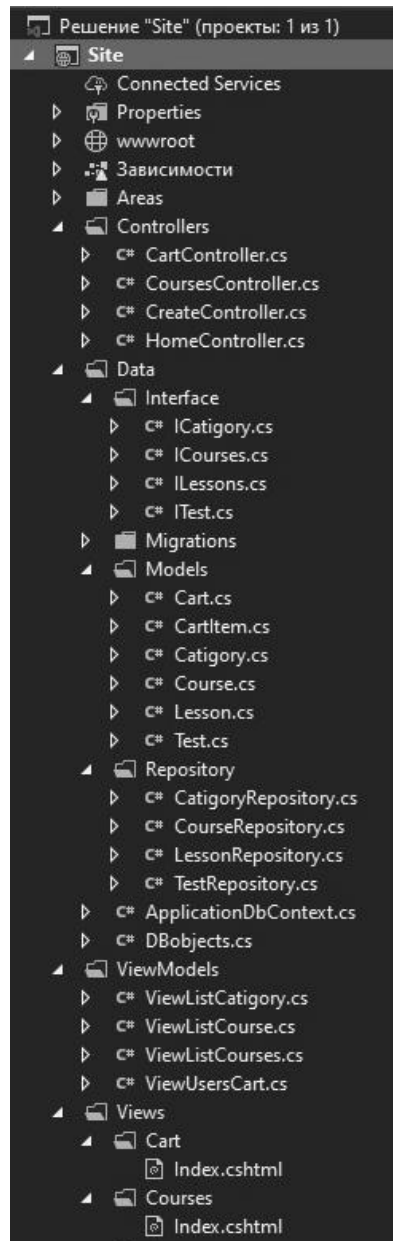


Рисунок 8 – Классы веб-приложения

Представленные на рисунке 9 представления созданы для отображения моделей с помощью контроллеров.

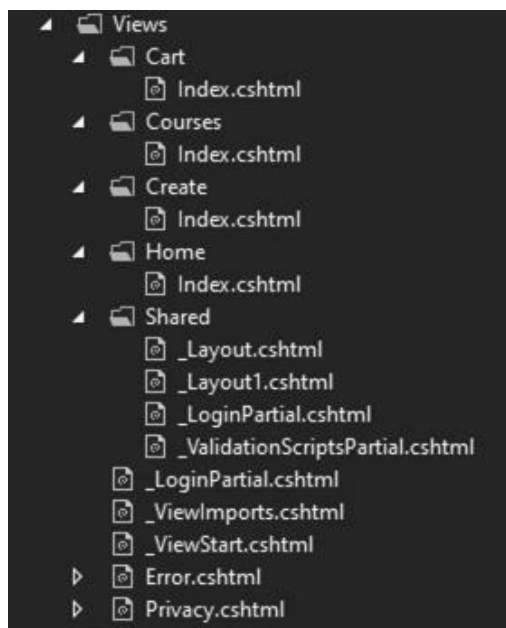


Рисунок 9 – Представления веб-приложения

1. /Controllers. В папку помещаются классы контроллеров.
2. /Data. В эту папку помещаются папки, содержащие модели, миграции, интерфейсы и репозитории.
3. /Models. Эта папка содержит классы моделей представлений и моделей предметной области.
4. /Interface. Здесь находятся интерфейсы для работы с моделями.
5. /Repository. Эта папка содержит репозитории, реализующие интерфейсы.
6. /ViewsModels. В эту папки помещены модели, содержащие в себе данные для передачи в представления через контроллер.
7. /Views. Здесь находятся представления.

3.1.2 Создание контекста Entity Framework

В проект мы установили пакет EntityFramework. Последние версии Entity Framework включают сначала код (code-first). В модели проекта можно определить классы, а затем на их основе сгенерировать базу данных. Сначала Entity Framework необходимо подключить к базе данных. Для этого добавляем в файл Web.config строки подключения к базе данных.

Создан класс `ApplicationDbContext`, содержащий в себе поля, на которых основана таблица. Класс `ApplicationDbContext` представлен в листинге 1.

Листинг 1 –Класс `ApplicationDbContext`

```
public class ApplicationDbContext : IdentityDbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }
    public DbSet<Lesson> Lessons { get; set; }
    public DbSet<Course> Courses { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<CartItem> CartItems { get; set; }
    public DbSet<Test> Tests { get; set; }
    public DbSet<UserMessage> Messages { get; set; }
}
```

Также настроили файлы `dbsettings.json`, `appsettings.json` и `startup.cs` для корректного создания и дальнейшей работы с базой данных. На листинге 2 представлена измененная часть код `startup.cs`.

Листинг 2 – Изменённый фрагмент кода `Startup.cs`

```
private IConfigurationRoot _confString;
public Startup(IHostingEnvironment hostEnv)
{
    _confString = new ConfigurationBuilder()
        .SetBasePath(hostEnv.ContentRootPath)
        .AddJsonFile("dbsettings.json").Build();
}

public IConfiguration Configuration { get; }
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options => options
        .UseSqlServer(_confString.GetConnectionString("DefaultConnection")));
}
```

```

services.AddDefaultIdentity<IdentityUser>(options => op-
tions.SignIn.RequireConfirmedAccount = true)
.AddEntityFrameworkStores<ApplicationDbContext>();
services.AddRazorPages();
services.AddTransient<ICategory, CategoryRepository>();
services.AddTransient<ICourses, CourseRepository>();
services.AddTransient<ILessons, LessonRepository>();
services.AddTransient<ITest, TestRepository>();
    services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();
services.AddReact();
services.AddScoped(sp => Cart.getCart(sp));
services.AddMvc(option => option.EnableEndpointRouting = false);
services.AddMemoryCache();
services.AddSession();
services.AddSignalR();
services.AddJsEngineSwitcher(options => options.DefaultEngineName =
ChakraCoreJsEngine.EngineName).AddChakraCore();
}

```

Затем при помощи консоли диспетчера пакетов создана миграция, и на её основе базу данных.

3.1.3 Создание базы данных

Для интерактивного обучающего сайта требовалось создание базы данных, в которой хранится информация о курсах, уроках, категориях, пользователях, пользовательских ролях. База данных сайта состоит из 7 таблиц и представлена на рисунке 10.

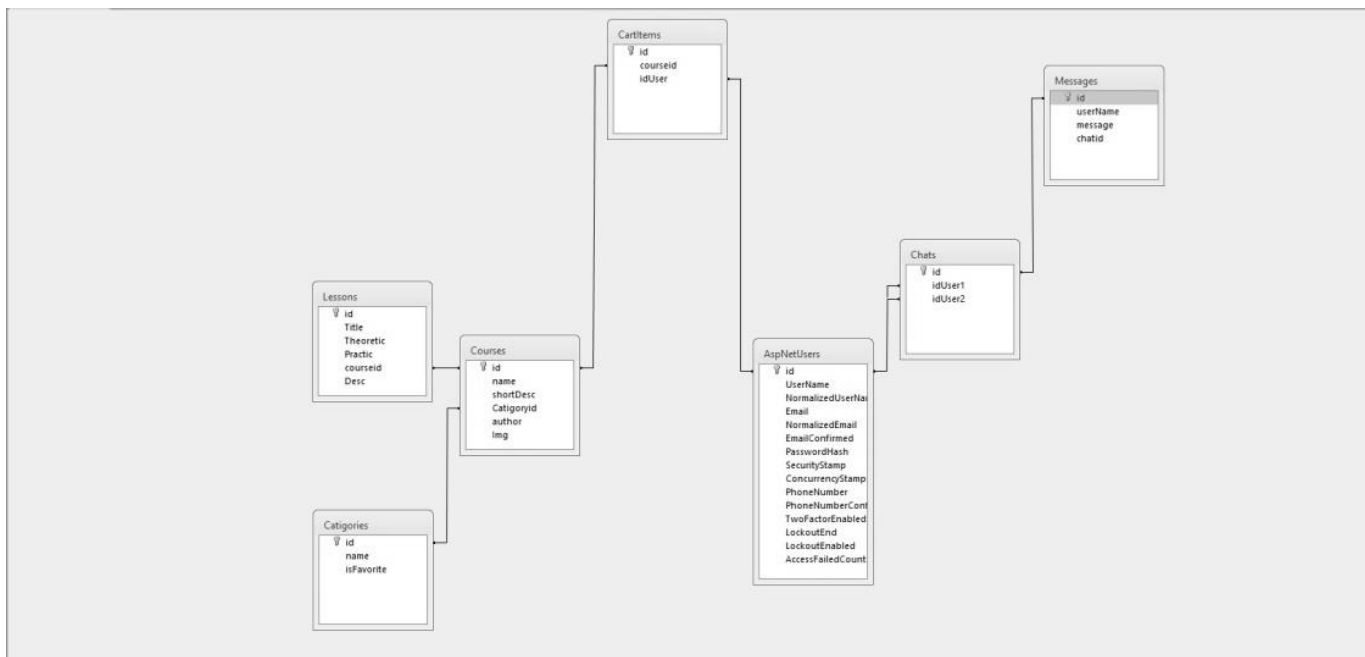


Рисунок 10 – Диаграмма базы данных

Полный перечень таблиц, созданных в базе данных для нашего интерактивного сайта для обучения, представлен на рисунке 11.

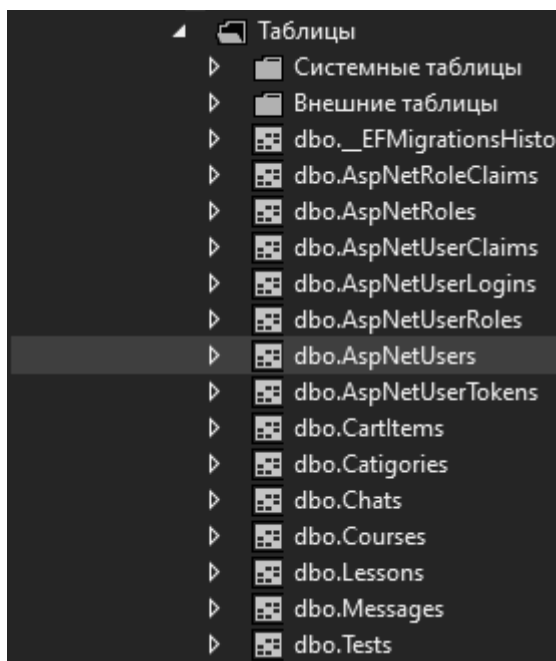


Рисунок 11 – База данных сайта

С помощью пакета Identity автоматически созданы `dbo._EFMigrationsHisto`, `dbo.AspNetRoleClaims`, `dbo.AspNetRoles`, `dbo.AspNetUserClaims`, `dbo.AspNetUserLogins`, `dbo.AspNetUserRoles`,

dbo.AspNetUser, dbo.AspNetUserTokens таблицы для хранения пользовательской информации для авторизации.

Таблица dbo.AspNetUser в базе данных содержит уникальный номер пользователя и его данные, предоставляемые при авторизации. Столбцы таблицы указаны на рисунке 12.

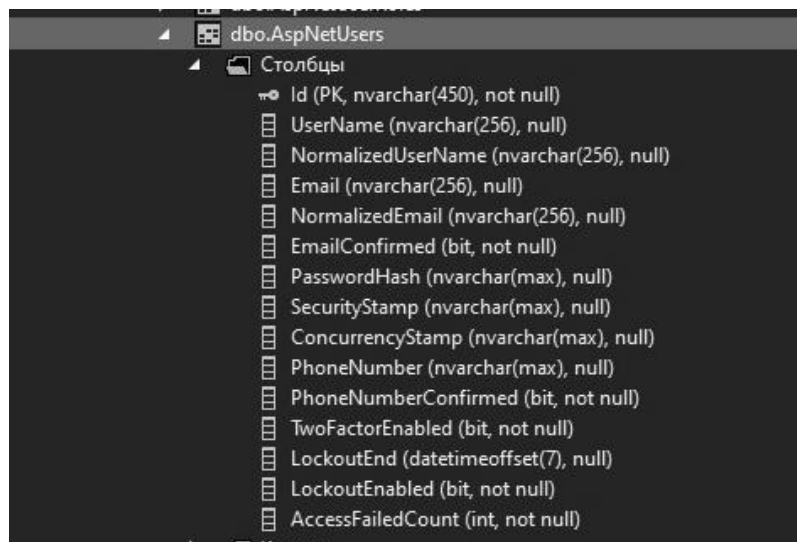


Рисунок 12 – Таблица dbo.AspNetUser

Таблица dbo.CartItems необходима для хранения данных о курсах, которых выбирает пользователь на сайте. Когда пользователь добавляет выбранный курс себе в личный кабинет, то автоматически происходит запись в данную таблицу. Заполненная таблица представлена на рисунке 13.

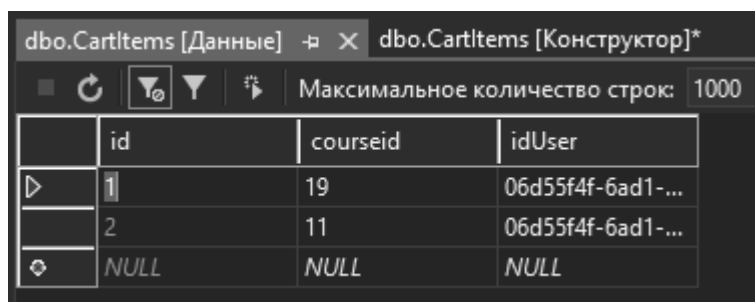


Рисунок 13 – Заполненная таблица dbo.CartItems

Таблица dbo.Catigories содержит данные о категориях курсов, представленных на сайте. В таблицу записывается id, name и isFavorite (статус категории). Заполненная таблица представлена на рисунке 14.

	id	name	isFavorite
	6	Информацион...	True
	7	Гуманитарные...	True
	8	Естественные ...	True
	9	Бизнес и мен...	False
	10	Статистика	False
	NULL	NULL	NULL

Рисунок 14 – Таблица dbo.Categories

Таблица dbo.Courses представляет собой таблицу курсов, который создает пользователь в режиме преподавателя. В таблице отображаются записи о категории, в которой находится курс, автор, который создает курс, id, наименование и краткое описание. Заполненная таблица представлена на рисунке 15.

	id	name	shortDesc	Categoryid	author	Img
	9	Основы статис...	В рамках курс...	5	IvanovIvan	/File/Course/03...
	10	HTML	Изучаем осно...	1	IvanovIvan	/File/Course/0h...
	11	Основы C#	C# — объектн...	1	PetrPetrov	/File/Course/01...
	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 15 – Таблица dbo. Courses

Таблица dbo.Lessons представляет собой таблицу уроков. В таблице отображаются записи о теме урока, описании, теории и практике, в которых отображаются ссылки на файлы для скачивания. Заполненная таблица представлена на рисунке 16.

	id	Title	Theoretic	Practic	courseid	Desc
	1	Общая инфор...	/File/Theoretic/...	/File/Practic/90...	9	Вводный ввод...
	2	Генеральная с...	/File/Theoretic/...	/File/Practic/90...	9	Урок направле...
	3	Основы HTML ...	/File/Theoretic/...	/File/Practic/10...	10	В этой главе м...
	4	HTML-теги	/File/Theoretic/...	/File/Practic/10...	10	В этом уроке ...
	5	Операторы inp...	/File/Theoretic/...	/File/Practic/11...	11	Оператор INP...
	6	Оператор outp...	/File/Theoretic/...	/File/Practic/11...	11	Знакомство с ...
	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 16 – Таблица dbo.Lessons

Для реализации чата на сайте между преподавателем и учеником созданы еще 2 таблицы. Таблица dbo.Messages, представленная на рисунке 17, содержит данные об id пользователя, который написал сообщение, автора сообщения и текст сообщения.

	id	userName	message
	1	PetrPetrov	Здравствуйте, ...
	NULL	NULL	NULL

Рисунок 17 – Таблица dbo. Messages

Таблица dbo.Chat отображает связь пользователей в чате. Заполненная таблица представлена на рисунке 18.

	id	idUser1	idUser2
	1	8aab4b5c-c3ae...	97b280f9-7db0-...
	NULL	NULL	NULL

Рисунок 18 – Таблица dbo.Chat

3.1.4 Разработка контроллеров

Для отображения данных из БД разработаны контроллеры.

Контроллер HomeController.cs был создан для передачи данных на страницу Home/Index и имеет поле ViewCategories для передачи всех категорий.

Контроллер `CoursesController.cs`, представленный в листинге 3, создан для передачи и обработки данных, отображаемых на странице подбора курсов. Имеет поле `ViewCourses` для передачи списка курсов и критерий подбора курсов.

Листинг 3 – `CoursesController.cs`

```
public class CoursesController : Controller
{
    private readonly ICatigory _catigory;
    private readonly ICourses _courses;
    private readonly ApplicationDbContext appDB;
    public CoursesController(ICourses _icourses, ICatigory _icatigory, ApplicationDbContext appDB)
    {
        _catigory = _icatigory;
        _courses = _icourses;
        this.appDB = appDB;
    }
    public async Task<IActionResult> Index(int idcatigories, string author)
    {
        var obj = new ViewListCourses
        {
            currCatigory = "",
            AllCourse = _courses.Courses,
            AllCatigories = _catigory.Catigories
        };
        if(idcatigories>0)
        {
            obj.currCatigory = _catigory.Catigories.FirstOrDefault(c => c.id == idcatigories).name;
            obj.AllCourse = obj.AllCourse.Where(c => c.Catigory.name == obj.currCatigory);
        }
        if (!String.IsNullOrEmpty(author))
        {
            obj.AllCourse = obj.AllCourse.Where(c => c.author == author);
        }
    }
}
```

```

        return View(obj);
    }
}

```

Для представления курсов выбранных пользователем разработан контроллер CartController.cs, включающий в себя функцию добавления курса в корзину и вывода информации обо всех выбранных курсах. В листинге 4 представлена часть кода, реализующая добавление в корзину.

Листинг 4 – Функция добавления в корзину addToCart()

```

public RedirectToActionResult addToCart(int id)
{
    var item = _icourse.Courses.FirstOrDefault(i => i.id == id);
    if (item != null)
    {
        _cart.idUser = User.Identity.GetUserId();
        _cart.addToCart(item);
    }
    return RedirectToAction("Index");
}

```

Реализация представления режима преподавателя осуществлена в контроллере CreateController.cs, состоящем из функций создания курса, создания урока и вывод курсов и уроков, созданных пользователем. В листинге 5 представлена функция сохранения курса.

Листинг 5 – функция сохранения курса SaveCourse

```

public async Task<RedirectToActionResult> SaveCourse(string catigory,
IFormFile cimg, Course course)
{
    if (cimg != null)
    {
        // путь к папке Files
        string path = "/File/Course/" + course.id + cimg.FileName;
        // сохраняем файл в папку Files в каталоге wwwroot
        using (var fileStream = new FileStream(_appEnvironment.WebRootPath
+ path, FileMode.Create))

```

```

        {
            await cimg.CopyToAsync(fileStream);
        }
        course.Img = path;
    }
    course.Categoryid = _icategory.Categories.FirstOrDefault(c => c.name
== category).id;
    course.author = User.Identity.Name;
    appDB.Courses.Add(course);
    appDB.SaveChanges();
    return RedirectToAction("Index");
}

```

Для реализации отображения чата для общения преподавателя с обучающимся разработан контроллер ChatController.cs. Рассматриваемый контроллер реализует вывод и сохранение сообщений написанных пользователями.

3.1.5 Подключение пакета Identity и настройка страниц авторизации

При создании проекта мы подключили пакет Identity, который реализует возможность пользователям создавать учетные записи и аутентифицироваться.

Для реализации страниц авторизации и регистрации разработаны модели с представлениями на основе класса Identity. Части модели авторизации реализующая аутентификацию пользователя представлена в листинге 6.

Листинг 6 – Функция OnPostAsync реализующая аутентификацию пользователя

```

public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl = returnUrl ?? Url.Content("~/");

    if (ModelState.IsValid)
    {
        var result = await _signInManager.PasswordSignInAsync(Input.Name,
Input.Password, Input.RememberMe, lockoutOnFailure: false);
        if (result.Succeeded)
        {

```

```

        _logger.LogInformation("User logged in.");
        return LocalRedirect(returnUrl);
    }
    if (result.RequiresTwoFactor)
    {
        return RedirectToPage("./LoginWith2fa", new { returnUrl = returnUrl, RememberMe = Input.RememberMe });
    }
    if (result.IsLockedOut)
    {
        _logger.LogWarning("User account locked out.");
        return RedirectToPage("./Lockout");
    }
    else
    {
        ModelState.AddModelError(string.Empty, "Invalid login attempt.");
        return Page();
    }
}
// If we got this far, something failed, redisplay form
return Page();
}
}

```

Для понятного и простого отображения страниц интерактивного сайта для обучения решено использовать компоновку, содержащую верхний заголовок, основное тело страницы и footer. Для предоставления стилей CSS, которые применены к приложению, мы воспользовались пакетом Bootstrap.

Twitter Bootstrap – набор инструментов для создания веб-приложений, включающий в себя шаблоны HTML и CSS оформления для различных элементов необходимых полноценному сайту и находящийся в открытом доступе. Сам по себе Bootstrap представляет набор файлов с готовыми стилями, накопленными студией разработки Twitter и выложенный в открытый доступ. В проект установлен пакет Bootstrap и дорабатывали шаблон CSS по своему усмотрению. Готовая начальная страница представлена на рисунке 25.

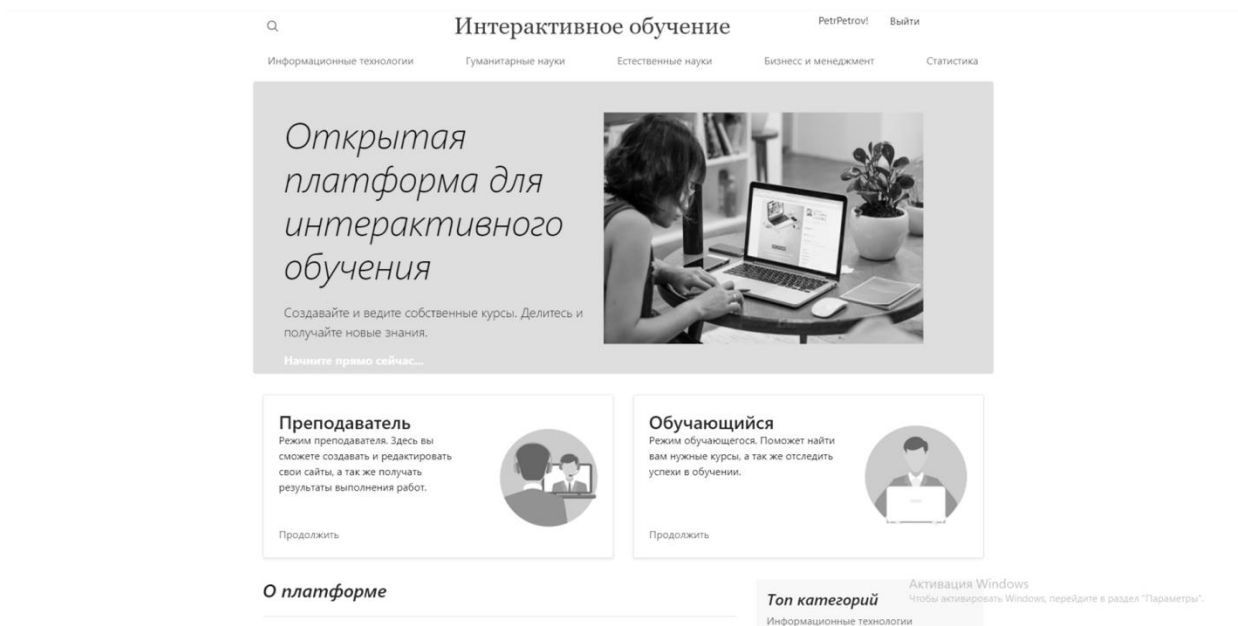


Рисунок 25 – Домашняя страница сайта

На главной странице в верхней панели отображаются категории, существующие на сайте. Код вывода представлен в листинге 7.

Листинг 7 – Вывод курсов

```
<div class="nav-scroller py-1 mb-2">
  <nav class="nav d-flex justify-content-between">
    @foreach (var catigory in Model.AllCatigories)
    {
      <a class="p-2 text-muted" asp-controller="Courses" asp-
action="Index" asp-route-idcatigories="@catigory.id">@catigory.name</a>
    }
  </nav>
</div>
```

Листинг 7 – Вывод курсов

Для удобства пользователей на главную страницу мы вывели топ курсов. Код вывода топа курсов представлен в листинге 8.

Листинг 8 – Вывод топа курсов

```
<div class="p-3 mb-4 bg-light rounded">
  <h4 class="font-italic">Топ категорий</h4>
  <ol class="list-unstyled mb-0">
    @foreach (var catigory in Model.AllCatigories)
```

```

        {
            if (category.isFavorite)
            {
                <p>
                    <a asp-controller="Courses" asp-action="Index"
asp-route-idcategories="@category.id">@category.name</a>
                </p>
            }
        }
    </ol>
</div>

```

На сайте реализовано два вида просмотра курсов: Режим обучающегося и режим преподавателя.

В режиме Преподавателя пользователи видят свои курсы, а также могут создавать новые. Часть кода реализующая сохранение урока представлена в листинге 9.

Листинг 9 – Функция SaveLesson реализующая сохранение урока.

```

public async Task<RedirectToActionResult> SaveLesson(string course, Lesson
lesson , IFormFile filePractic, IFormFile fileTheprectic)
{
    if (filePractic != null)
    {
        // путь к папке Files
        string path = "/File/Practic/" + _icourse.Courses.FirstOrDefault(c
=> c.name == course).id + lesson.id+filePractic.FileName;
        // сохраняем файл в папку Files в каталоге wwwroot
        using (var fileStream = new FileStream(_appEnvironment.WebRootPath
+ path, FileMode.Create)) {
            await filePractic.CopyToAsync(fileStream);
        }
        lesson.Practic = path;
    }
    if (fileTheprectic != null)
    {
        // путь к папке Files

```



```

        string path = "/File/Theoretic/" +
        _icourse.Courses.FirstOrDefault(c => c.name == course).id + lesson.id +
        fileTheprectic.FileName;

        // сохраняем файл в папку Files в каталоге wwwroot
        using (var fileStream = new FileStream(_appEnvironment.WebRootPath
+ path, FileMode.Create)) {
            await fileTheprectic.CopyToAsync(fileStream);
        }
        lesson.Theoretic = path;
    }
    lesson.course = _icourse.Courses.FirstOrDefault(c => c.name ==
course);

    appDB.Lessons.Add(lesson);
    appDB.SaveChanges();
    return RedirectToAction("Index");
}

```

Для работы в режиме Пользователя создана корзина, в которую пользователи могут добавлять понравившиеся курсы. В листинге 10 представлены основная функция для передачи и обработки информации из базы данных в режим Обучающегося.

Листинг 10 – Основная функции Index()

```

public ActionResult Index()
{
    _cart.idUser = User.Identity.GetUserId();
    var cart = _cart.getItems();
    _cart.ListCartItem = cart;
    var lessons = _ilesson.Lessons;
    var test = _itest.tests;
    var obj = new ViewUsersCart
    {
        Cart = _cart,
        AllLesson = lessons,
        AllTest = test
    };
    return View(obj);
}

```

```
}
```

Для упрощения поиска курсов на сайте реализован поиск и отбор по категориям и автору.

Для этого реализована форма отправки данных в класс контроллера и обработки переданной модели. В листинге 11 представлена страница поиска курсов.

Листинг 11 – Часть кода страницы поиска курсов реализующая вывод курсов

```
<div class="row">
    @{
        foreach (var course in Model.AllCourse)
        {
            <div class="col-lg-4 mt-3">
                

                <h2>@course.name</h2>
                <p>@course.shortDesc</p>
                <p>@course.author</p>
                <p><a
                    class="btn
                    btn-secondary"
href="~/Cart/addToCart/@course.id" role="button">Добавить</a></p>
            </div><!-- /.col-lg-4 -->
        }
    }
</div>
```

Для реализации общения в реальном времени между пользователем и преподавателем подключен пакет SignalR.

SignalR – это бесплатная библиотека программного обеспечения с открытым исходным кодом для Microsoft ASP.NET, которая позволяет серверному коду отправлять асинхронные уведомления клиентским веб-приложениям. Библиотека включает компоненты JavaScript на стороне сервера и на стороне клиента [10].

Разработан скрипт, помогающий обработать данные полученные с формы. Код рассматриваемого скрипта представлен в листинге 12.

Листинг 12 – Функции для обработки данных с HTML-формы

```

"use strict";

var connection = new signalR.HubConnectionBuilder().withUrl("/chatHub").build();

//Disable send button until connection is established
document.getElementById("sendButton").disabled = true;

connection.on("ReceiveMessage", function (user, message) {
    var msg = message.replace(/&/g, "&amp;").replace(/</g, "&lt;").replace(/>/g,
"&gt;");
    var encodedMsg = user + " says " + msg;
    var li = document.createElement("li");
    li.textContent = encodedMsg;
    document.getElementById("messagesList").appendChild(li);
});

connection.start().then(function () {
    document.getElementById("sendButton").disabled = false;
}).catch(function (err) {
    return console.error(err.toString());
});

document.getElementById("sendButton").addEventListener("click", function (event) {
    var user = document.getElementById("userInput").value;
    var message = document.getElementById("messageInput").value;
    var file = document.getElementById("messageFile").value;
    connection.invoke("SendMessage", user, message, file).catch(function (err) {
        return console.error(err.toString());
    });
    event.preventDefault();
});

```

3.2 Руководство пользователя

3.2.1 Главная страница сайта

При открытии интерактивного сайта для дистанционного обучения пользователь видит начальную страницу, на которой в шапке доступна

регистрация или аутентификация в веб-приложении. Шапка начальной страницы сайта приведена на рисунке 26.



Рисунок 26 – Шапка начальной страницы

При выборе соответственно входа или регистрации пользователю открывается форма для заполнения своих данных. Форма регистрации представлена на рисунке 27. Необходимо ввести E-mail, имя пароль.

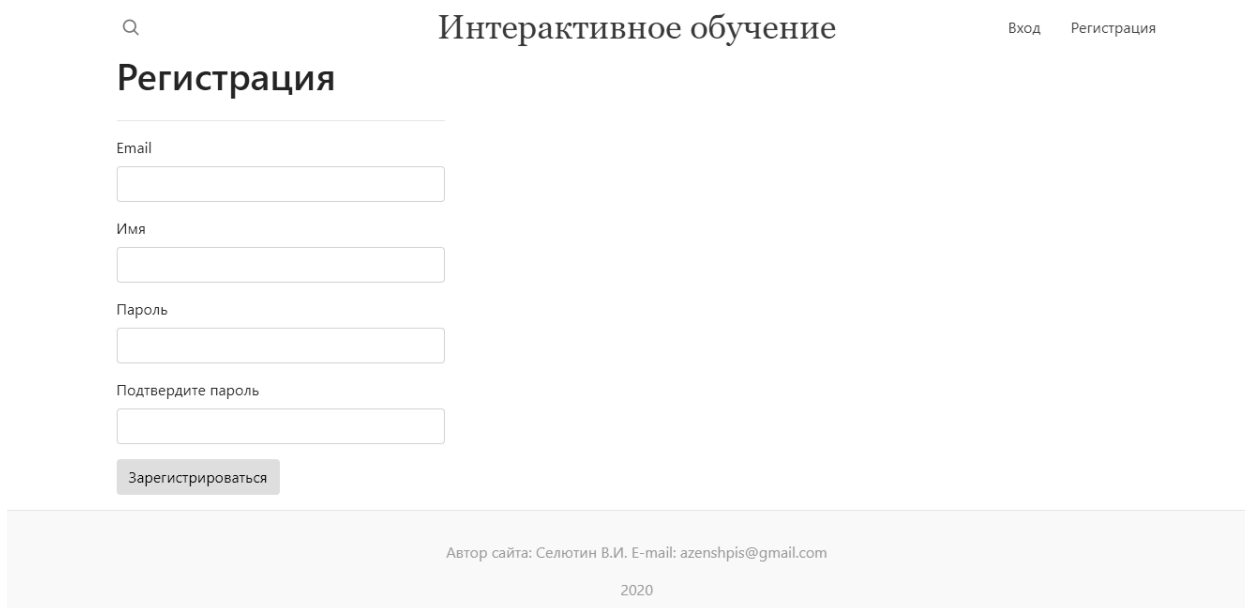


Рисунок 27 – Форма регистрации

Под шапкой расположен список доступных на сайте категорий курсов, при нажатии на название категории пользователь перемещается на страницу с курсами по выбранной категории. Список категорий приведен на рисунке 28.



Рисунок 28 – Категории веб-приложения

Под списком категорий расположен информационный блок содержащий ссылку на страницу поиска курсов. Информационный блок с выделенной ссылкой приведен на рисунке 29.



Рисунок 29 – Кнопка перехода на страницу поиска курсов по категории и автору

При нажатии на кнопку пользователь переходит на страницу поиска курсов. Под этим блоком на главной странице располагается информационная часть страницы. Основная её часть находится слева и содержит в себе краткий перечень возможностей сайта.

Справа в информационной части доступен список топ категорий ссылками на них и контакты. Правая часть информационной части главной страницы приведена на рисунке 30.

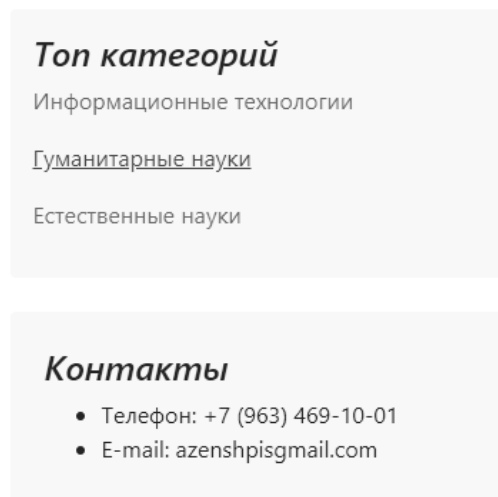


Рисунок 30 – Правая часть информационной части главной страницы

В футере сайта располагается информация об авторе сайта и годе его создания.

3.2.2 Страница подбора курсов

Страница подбора курсов, представленная на рисунке 31, содержит в себе полный перечень курсов доступных на сайте. В центре страницы доступен выбор фильтров для более быстрого поиска нужного курса. На сайте доступен выбор категории и автора, на странице отобразится отбор по выбранным критериям.



Рисунок 31 – Страница отбора курсов по критериям

При выборе курса пользователь попадает на страницу режима обучающегося, а курс автоматически добавляется в корзину.

3.2.3 Режим обучающегося

На рисунке 32 приведён режим Обучающегося позволяющий пользователю просматривать ранее выбранные курсы. Страница состоит из левой панели с выбором курсов и уроков и главной части страницы отображающей выбранный урок.

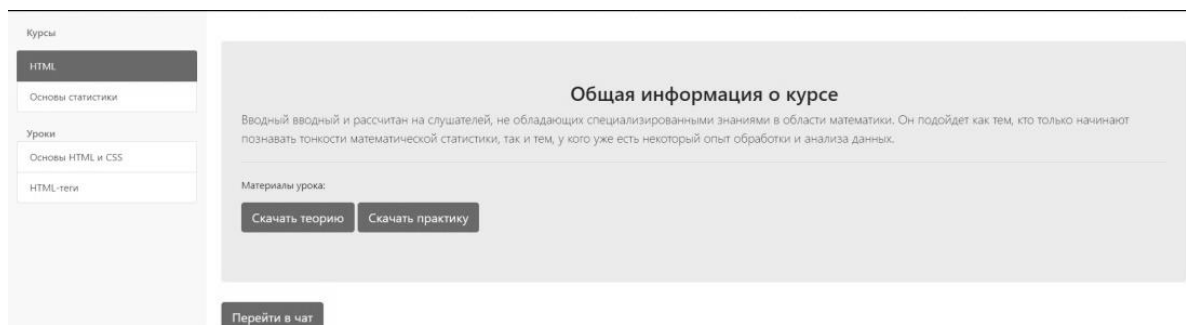


Рисунок 32 – Корзина обучающегося с выбранными курсами

При выборе урока в главной части страницы отображается тема и краткое описание, либо указания к уроку. Обучающийся может, при нажатии на кнопку скачать теорию – ознакомиться с теорией, а соответственно нажав кнопку практики – ознакомиться с практикой.

Для связи с преподавателем необходимо нажать на кнопку «Перейти в чат». Откроется страница с диалогом между преподавателем и обучающимся, который в режиме реального времени отображает сообщения, написанные в чате. Страница чата приведена на рисунке 33.

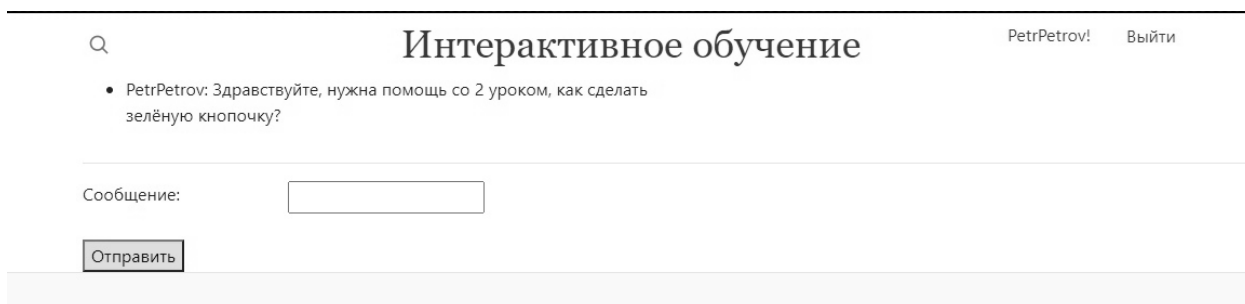


Рисунок 33 – Страница чата

3.2.4 Режим преподавателя

Режим преподавателя позволяет создавать и управлять собственными курсами пользователя. При переходе в этот режим на левой панели страницы отображаются курсы, автором которых является пользователь. Так же доступны 2 кнопки для добавления курса и добавления урока.

При нажатии кнопки добавления курса, которая расположена вверху левой части страницы и выглядит в форме «+», на главной части страницы отображается форма добавления курса, состоящая из выбора категории, названия курса, загрузки картинки к курсу, кратким описанием и кнопкой сохранения. Форма создания курса реализована на рисунке 34.

Рисунок 34 – Форма создания курса преподавателем

Затем преподавателю необходимо создать уроки. Создание урока доступно по аналогичной кнопке ниже. При создании урока необходимо указать имя, краткое описание и загрузить теоретический и практический файлы для обучающегося. Форма создания урока приведена на рисунке 35.

Рисунок 35 – Форма создания урока преподавателем

Выводы по разделу три

В третьей главе выпускной квалификационной работы разработан интерактивный сайт для дистанционного обучения. Подробно описан процесс разработки веб-приложения, страницы и элементы которого созданы специально для организации процесса дистанционного обучения. Разработан понятный интерфейс сайта для пользователей.

В отдельном пункте главы написано краткое руководство пользователя, в котором изложено, как пользователи должны работать с разработанным сайтом, а также приведен экономический анализ проекта.

4 ЭКОНОМИЧЕСКИЙ АНАЛИЗ ПРОЕКТА

Разработка программного продукта производилась по инициативе разработчика. В связи с этим, финансирование проекта реализовано за его счет.

В расчётах стоимости разработки сайта основными затратами являются:

- зарплата веб-разработчика;
- услуги размещения на хостинге;
- покупка доменного имени.

Данные о зарплате веб-разработчика взяты с сайта <https://chelyabinsk.hh.ru/>. HeadHunter – крупнейшая российская компания интернет-рекрутмента. Мы взяли среднее значение о заработной плате зарплате веб-разработчика: 40000руб./месяц. Пример вакансии веб-разработчика приведен на рисунке 36.

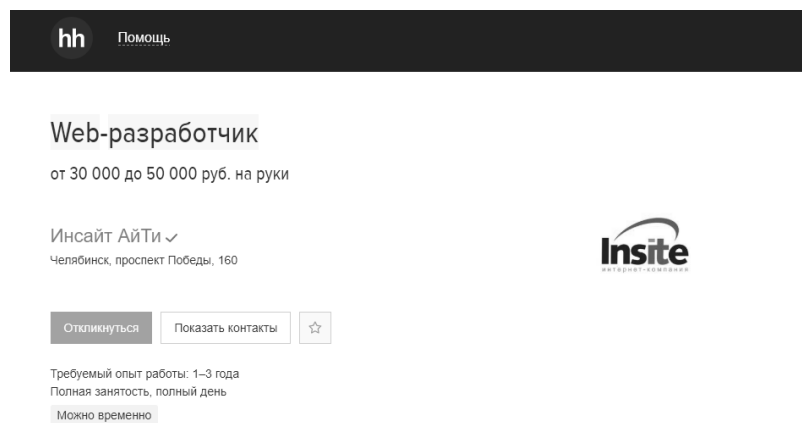


Рисунок 36 – Пример вакансии веб-разработчика

Данные по стоимости услуг веб-хостинга взяты с сайта <https://www.reg.ru/>. Так же в услуги хостинга входит и оплата домена. На рисунке 37 представлена услуга размещения сайта на веб-хостинг.

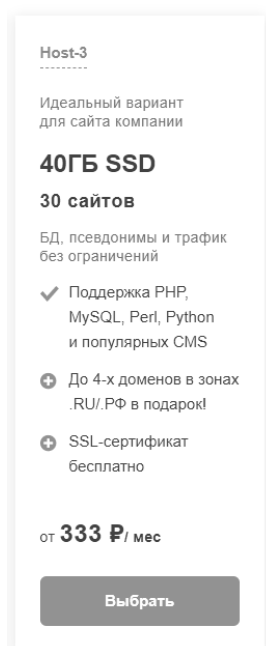


Рисунок 37 – Стоимости услуг веб-хостинга

В таблице 2 приведены сводные данные о затратах на разработку и развёртывание сайта. А так же общая сумма с учетом оплаты услуг хостинга на год.

Таблица 2 – Сводные данные о затратах на разработку сайта

Услуга	Стоимость	Количество	Сумма, руб
Услуги веб-разработчика	40000руб./месяц	1	40000 руб.
Услуги веб-хостинга	333руб./месяц	12	3996руб.
Оплата доменного имени	0руб./месяц	12	0руб.
Итого			43996 руб.

В экономическом разделе рассчитана стоимость создания веб-приложения и размещение его на веб-хостинге в течение года.

Вывод по разделу четыре

В четвертой главе выпускной квалификационной работы проведён экономический анализ разработанного сайта для интерактивного обучения, а на основании актуальных данных была рассчитана себестоимость проекта

ЗАКЛЮЧЕНИЕ

Выпускная квалификационная работа по разработке интерактивного сайта для дистанционного обучения, позволяющая создавать собственные курсы или получать новые знания, позволит предоставлять познавательный контент, а также поспособствует в приобретении необходимых навыков онлайн, на обучающей платформе.

В результате проделанной работы решены следующие задачи:

- проведен анализ предметной области;
- спроектирована информационная модель объекта;
- разработан сайт для интерактивного обучения.

Интерактивный сайт для обучения объединяет в себе весь функционал, необходимый для организации онлайн обучения. А благодаря своей гибкости и универсальности, возможно дальнейшее продвижение интерактивного сайта для обучения с целью получения экономической выгоды.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Беэр Бибо, Иегуда Кац. jQuery. Подробное руководство по продвинутому JavaScript / Беэр Бибо, Иегуда Кац. – Символ-Плюс, 2011. – 624 с.
2. Адам Фримен. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов. / Адам Фримен. – Вильямс, 2018. – 736 с.
3. Мэтью Мак-Дональд, Марио Шпушта. Microsoft ASP.NET 4.0 с примерами на C# 2010 для профессионалов, 4е издание – Вильямс, 2011. – 1424 с.
4. Стивен Хольцнер «Аjax Библия программиста» – Диалектика. Москва. СанктПетербург. Киев, 2009г – 560 с.
5. Основы создания Web-сайтов на C# в Visual Studio – http://wladm.narod.ru/C_Sharp/webbegin.html.
6. Начало работы с ASP.NET MVC 5 – <http://metanit.com/sharp/mvc5/1.2.php>.
7. Microsoft Visual Studio – https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio.
8. «SignalR» – <https://en.wikipedia.org/wiki/SignalR>.
9. «DFD» – описание нотации – <https://ru.wikipedia.org/wiki/DFD>.