МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра ____физики открытых систем_____
наименование кафедры

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ
Разработка веб-ресурса научных публикаций

Студента 4 курса 4041 группы

направления 09.03.02. Информационные системы и технологии

код и наименование направления

института физики

наименование факультета

Арустамяна Макара Романовича

фамилия, имя, отчество

Научный руководитель

профессор кафедры ФОС, д.ф.-м.н., доцент		О.И. Москаленко
должность, ученая степень, уч. звание.	личная подпись, дата	инициалы, фамилия

Зав. кафедрой

д.ф.-м.н., профессор		А.А. Короновский

Саратов 2022 г.

Введение

Публикация – это не только распространение данных о научных исследованиях и их верификация, но и также, по сути своей, самая главная мера эффективности научной работы. Для людей, не знакомых с научный процессом, важность научных публикаций может быть не очень понятна. Но на самом деле именно наукометрические показатели являются основой для получения грантов, карьерного роста ученых и самого научного успеха. Поэтому проблемы публикаций негативно влияют на саму науку в целом.

На сегодняшний день существует два способа «монетизации» научного журнала: подписка и открытый доступ. В первом случае можно купить возможность читать отдельную статью или оформить подписку на журнал в целом. Открытый доступ же позволяет читать журнал из любой точки мира и позволяет скачивать статьи из интернета.

Поговорим подробнее о модели платного доступа. Сейчас многие научные журналы требуют подписку для получения доступа к статье. Стоимость отдельно взятой статьи может доходить до десятков долларов, а каждому сотруднику нужен доступ к сотням публикаций, издатели договариваются с научными и образовательными учреждениями о подписке на большой пакет их публикаций на длительный срок. В этом случае сотрудники могут получить доступ ко всем публикациям издательства. Но даже в таком случае подписка обходится достаточно дорого. Сколько именно средств уходит на оплату подписки, неизвестно, издатели не разглашают эти данные. Но вузы многих стран, в том числе, например, Германии, отказываются от подписки из-за ее дороговизны. Основная же проблема заключается в том, что ученые не имеют доступа к статьям.

С развитием интернета, появились журналы, которые выходят только в сети. С появлением подобных ресурсов изменилась и модель «монетизации». В таком случае за статью платит тот, кто ее публикует, то есть сам исследователь. Основным преимуществом является то, что статьи находятся в открытом

доступе. Все желающие могут прочесть статью без каких-либо препятствий. Поэтому многие поддерживают переход на новую модель «монетизации». Однако, и в данной системе институты, а в ряде случаев и сами авторы, тратят огромные средства.

Целью данной бакалаврской работы является создание веб-ресурса, который будет предоставлять неограниченный доступ пользователям ко всем загруженным публикациям.

Для разработки подобного ресурса нужны технологии, позволяющие выдерживать большую нагрузку, которая может возникнуть в случае, если сайтом будет пользоваться большое количество пользователей одновременно.

ASP.NET – это серверная среда веб-приложений с открытым исходным кодом, созданная Microsoft, которая работает в Windows и была запущена в начале 2000-х годов. ASP.NET позволяет разработчикам создавать веб-приложения, веб-службы и веб-сайты с динамическим содержимым.

Для разработки под ASP.NET могут использоваться различны IDE. Если разработка ведется на операционной системе Windows, то разработку можно вести в программе Visual Studio.

Основная часть

Основная часть бакалаврской работы состоит из четырех разделов. В первом разделе описан процесс создания базы данных. Была разработана реляционная база данных проекта. Реляционные базы данных используют, как правило, язык запросов SQL. Структура такой базы данных представляет собой таблицы, поля которых описывают определенную сущность. Такая структура позволяет связывать между собой информацию из разных таблиц с помощью внешних ключей, которые используются для уникальной идентификации любого атомарного фрагмента данных в этой таблице. Другие таблицы могут ссылаться на этот внешний ключ, чтобы создать связь между частями данных и частью, на которую указывает внешний ключ.

Для создания локального сервера и базы данных использовалась программа SQL Server Management Studio. Эта утилита служит для управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

После установки данной программы, создается локальный сервер, на котором в дальнейшем и будет работать веб-ресурс.

Взаимодействие с базой данных происходит с помощью технологии Entity Framework Core. Entity Framework представляет технологию объектно-реляционного сопоставления (ORM) Microsoft для доступа к данным.

Взаимодействие с базой данных в Entity Framework Core происходит посредством специального класса – контекста данных. Поэтому в проект был добавлен новый класс, который назван ApplicationContext.

Следующим этапом разработки ресурса, описанным во втором разделе, была разработка клиентской части. Для этого использовался стандартный стек технологий front-end, язык гипертекстовой разметки HTML, язык описания внешнего вида документа CSS и язык программирования JavaScript.

Разработка велась в рамках архитектуры MVC (Model-View-Controller). Эта концепция предполагает разделение приложения на три компонента, как понятно из названия. Основная цель паттерна состоит в разделении бизнес-логики и данных от визуализации. Подобная архитектура упрощает сопровождение программного кода: правки в клиентской части не отражаются на бизнес логике.

Был создан метод в контроллере. Имя метода должно совпадать с именем файла представления. Метод получает на вход объект класса RegistryViewModel, в который записаны данные, заполненные пользователем в форме регистрации. Получая данные из представления, метод записывает их в базу данных. Также метод сохраняет данные пользователя в кэш, чтобы пользователю не приходилось заново авторизоваться при каждом посещении сайта.

Представление в ASP.NET MVC может содержать не только стандартный html-код, но и вставки кода C#. Механизм просмотра используется для обработки кода, который содержит как элементы html, так и конструкции C#.

По умолчанию ASP.NET MVC Core использует один механизм представления Razor, хотя, при желании, можно также использовать какие-то другие сторонние движки или самим создавать свой движок просмотра.

Цель механизма представления Razor – определить переход от разметки html к коду C#.

Следующим этапом разработки (раздел 3) является реализация возможности авторизоваться под своими учетными данными. Задача состоит в том, чтобы сравнивать данные, полученные от пользователя, с данными сервера.

Для реализации подобного функционала был создан метод в контроллере. На вход метод получает объект класса AuthorizationViewModel. Сам класс содержит такие поля как PhoneNumber, Password, OneTimePassword и прочие. Для отправки почты в среде интернет используется протокол SMTP (Simple Mail Transfer Protocol). Данный протокол указывает, как почтовые сервера взаимодействуют при передаче электронной почты.

Для работы с протоколом SMTP и отправки электронной почты в .NET предназначен класс SmtpClient из пространства имен System.Net.Mail.

Также существует возможность блокировать пользователей. Если пользователь заблокирован, появляется модальное окно с ошибкой. Следующим этапом идет отправка сообщения на почту или номер телефона, который пользователь указал при регистрации. Код отправляется на номер телефона пользователя в случае, если он пользуется русскоязычным интерфейсом, в противном случае код отправляется на почту пользователя. После верного введения кода пользователь переадресовывается на страницу личного кабинета, который подробно описан в разделе 4 бакалаврской работы.

Личный кабинет создан для того, чтобы пользователь мог редактировать личную информацию. Была создана страница с формой, ее вид изображен на рисунке 1.

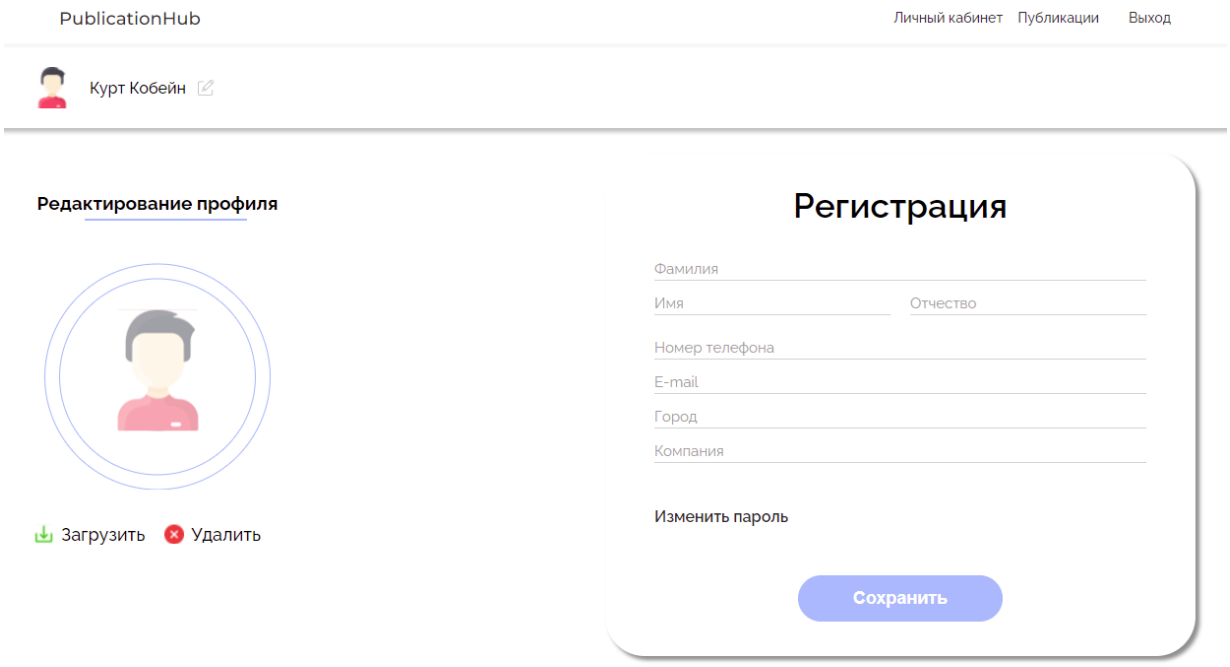

Рисунок 1 – Личный кабинет

Метод для обновления данных представлен на рисунке 2. Поскольку пользователь авторизован, его данные берутся из кэша браузера, после чего поля объекта заменяются новыми значениями из формы и записываются в базу данных.

```
Ссылок: 0
public async Task<IActionResult> Update(ProfileModel model)
{
    var user = UserModel.FromEntity(await new UsersBL().GetAsync((User.Identity as CustomUserIdentity).Id.Value));
    var res = await new UniversityBL().GetAsync(model.UniversityId.Value);
    if (ModelState.IsValid)
    {
        user.Firstname = model.Firstname;
        user.Lastname = model.Lastname;
        user.Email = model.Email;
        user.Patronymic = model.Patronymic;
        user.City = model.City;
        user.PhoneNumber = model.PhoneNumber;
        UniversityModel university;
        if (res == null)
        {
            university = new UniversityModel();
            university.Name = model.RestaurantName;
            university.Id = await new UniversityBL().AddOrUpdateAsync(UniversityModel.ToEntity(university));
            user.RestaurantId = (int?)university.Id;
        }
        else
        {
            university = UniversityModel.FromEntity(await new UniversityBL().GetAsync(model.UniversityId.Value));

            if (university == null)
            {
                ViewBag.Error = "Организация не найдена";
                return View(model);
            }
        }
        user.UniversityId = university.Id;

        await new UsersBL().AddOrUpdateAsync(UserModel.ToEntity(user));
    }
    return RedirectToAction("Index","Profile", new { Area = "Public"});
}
```

Рисунок 2 – Метод Update

Для загрузки публикаций была создана форма и соответствующий метод в контроллере. На рисунке 3 изображена страница просмотра публикаций.

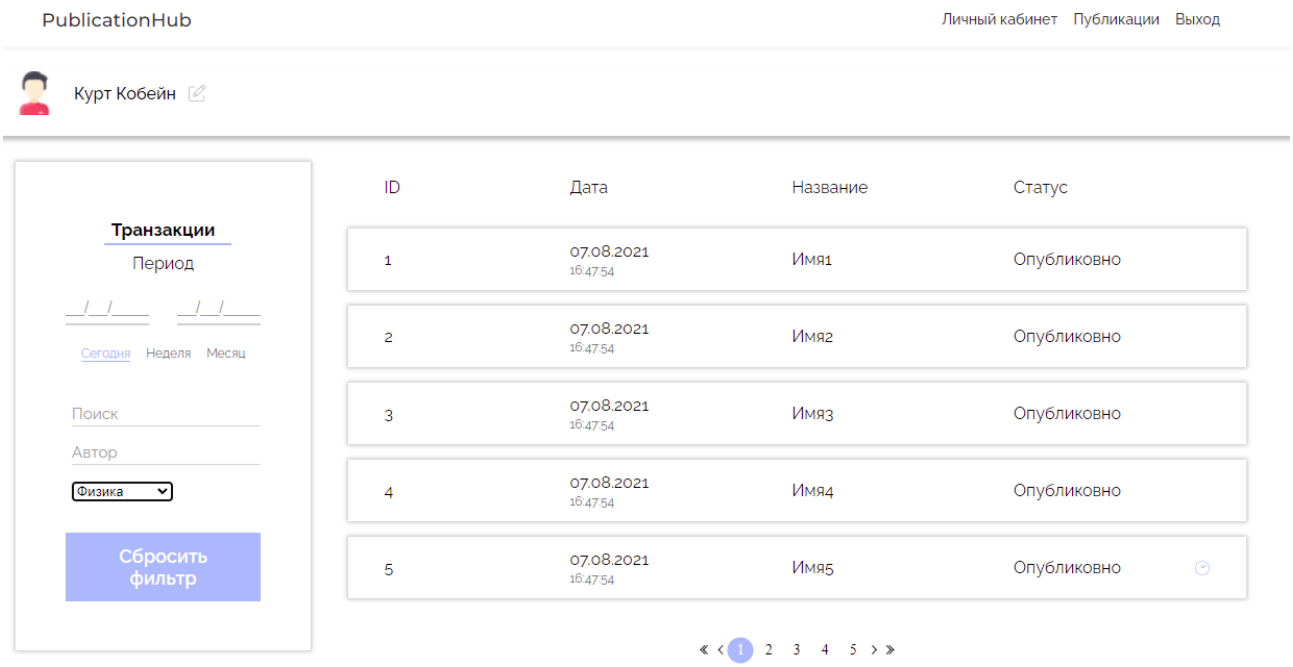

Рисунок 3 – просмотр публикаций в личном кабинете

Данный метод получает файл методом POST. Файл сохраняется на сервере, а в базу данных записывается относительный путь.

На рисунке 4 представлена форма загрузки публикации.

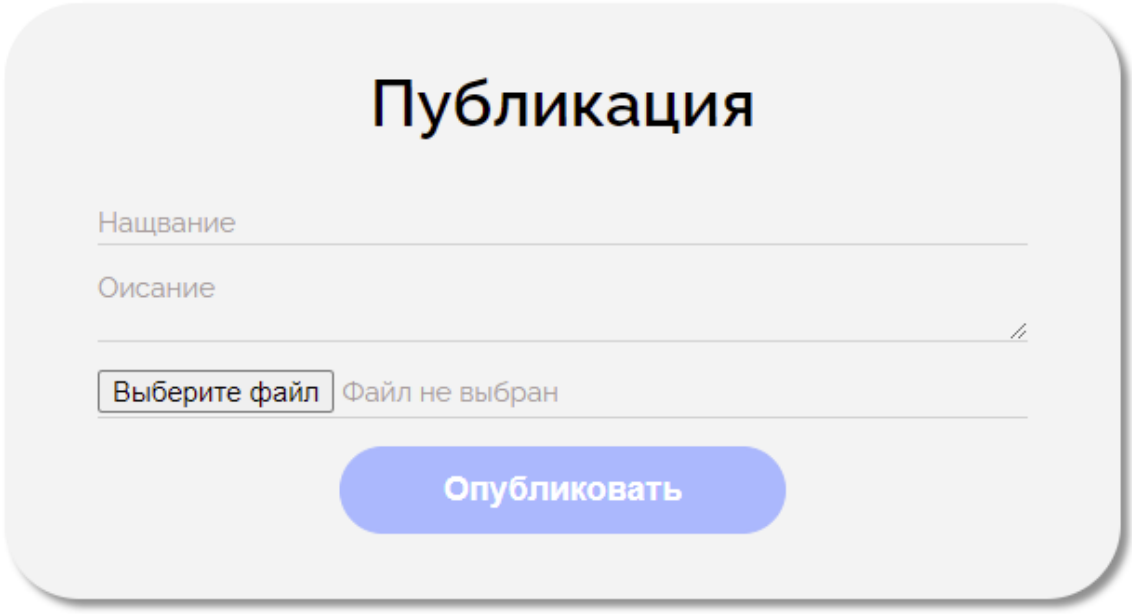

Рисунок 4 – Форма публикации

Таким образом получается, что pdf файла загружается в папку Files на сервер, в базе данных же сохраняется запись пути к файлу, а также остальные данные введенные с формы.

При нажатии на определенную публикацию открывается новое окно с соответствующим файлом в формате pdf.

Также пользователям доступен фильтр по датам и категориям для более удобного поиска материала.

Заключение

В рамках бакалаврской работы был разработан веб-сайт, который служит архивом публикаций. Он позволяет пользователям иметь неограниченный доступ ко всем загруженным материалам. В ходе работы был реализован весь необходимый функционал, хотя многое еще можно расширить и дополнить. Разработка велась на высокоуровневом языке программирования C#. Также использовалась технология создания веб-приложений ASP.NET. Проект разрабатывался с учетом будущей модификации и дальнейшего расширения, этому способствовала и архитектура MVC. Были изучены актуальные технологии и правила проектирования подобного рода ресурсов.

Список литературы

1. **“CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд.”** [Книга]
2. **“Руководство работы с ASP.NET Core”,** [Интернет ресурс]. URL: https://metanit.com/sharp/aspnet5/1.2.php
3. **“Паттерны/шаблоны проектирования”,** [Интернет ресурс]. URL: refactoring.guru
4. **“Работа с данными в Entity Framework”,** [Интернет ресурс]. URL: https://metanit.com/sharp/aspnet5/12.1.php
5. **“Основы HTML”,** [Интернет ресурс]. URL:https://html5book.ru/javascript-jquery/
6. **“Основы JavaScript”,** [Интернет ресурс]. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics
7. Документация microsoft по технологиям ASP.NET, Entity Framework, C#. [Интернет ресурс]. URL: https://docs.microsoft.com/