



# ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

*«Приднестровский государственный университет им Т. Г. Шевченко»*

ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра Бизнес-информатики и информационных технологий

*Допустить к защите*

И. о. зав. кафедрой

\_\_\_\_\_ Л.Ю. Надькин

«\_\_\_» \_\_\_\_\_ 2022 г.

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**Разработка веб ресурса для обеспечения деятельности**

**издательства ПГУ им. Т.Г. Шевченко**

Направление подготовки

38.03.05 Бизнес-информатика

Профиль

Электронный бизнес

Выполнила

студентка 401 группы

\_\_\_\_\_ Слободян Я.С.

Научный руководитель

Доцент

\_\_\_\_\_ Надькин Л.Ю.

Тирасполь, 2022

|  |           |
|--|-----------|
| <b>ОГЛАВЛЕНИЕ</b>  |           |
| <b>ВВЕДЕНИЕ .....</b>  | <b>3</b>  |
| <b>ГЛАВА 1. Теоретические исследования по издательской деятельности .....</b>                                | <b>5</b>  |
| <b>1.1 Издательство, типография, основные понятия .....</b>  | <b>5</b>  |
| <b>1.2 Анализ существующих веб ресурсов для типографии и издательской деятельности.....</b>                  | <b>6</b>  |
| <b>1.3 Проект разрабатываемого программно-технического решения .....</b>                                     | <b>7</b>  |
| <b>ГЛАВА 2. Проектирование веб ресурса для издательства ПГУ им. Т.Г. Шевченко.....</b>                       | <b>12</b> |
| <b>2.1 Краткая характеристика издательства ПГУ им. Т.Г. Шевченко.....</b>                                    | <b>12</b> |
| <b>2.2 Анализ имеющейся системы взаимодействия в издательстве и необходимость ее совершенствования .....</b> | <b>13</b> |
| <b>2.3 Структура и содержание веб ресурса.....</b>   | <b>16</b> |
| <b>2.3.1 Логическая структура ресурса.....</b>   | <b>16</b> |
| <b>2.3.2 Параметры и внешний вид ресурса .....</b>   | <b>18</b> |
| <b>2.3.3 Основные продукты для разработки .....</b>  | <b>20</b> |
| <b>ГЛАВА 3. Создание веб ресурса для издательства ПГУ им. Т.Г. Шевченко .....</b>                            | <b>25</b> |
| <b>3.1 Разработка веб ресурса .....</b>  | <b>25</b> |
| <b>3.2 Тестирование веб ресурса .....</b>  | <b>27</b> |
| <b>3.3 Расчет экономической эффективности веб ресурса .....</b>  | <b>33</b> |
| <b>ЗАКЛЮЧЕНИЕ .....</b>  | <b>40</b> |
| <b>СПИСОК ЛИТЕРАТУРЫ.....</b>  | <b>41</b> |
| <b>ПРИЛОЖЕНИЕ.....</b>   | <b>44</b> |

## ВВЕДЕНИЕ

Университетское книгоиздание реализуется в тесном взаимодействии с учебной и научной деятельностью, развитие и перспектива которых во многом определяется возможностями вузовской полиграфии.

В научной сфере речь идёт о стратегиях роста публикационной активности и улучшении аккредитационных и отчётных показателей вуза.

Формируется комплекс представлений о новом, достаточно широком спектре инструментария для роста эффективности вузовской науки и повышения её престижа на общегосударственном уровне. [7]

**Актуальность** темы исследования обусловлена увеличением уровня конкурентно способности ПГУ им. Т.Г. Шевченко, а также улучшением аккредитационных и отчётных показателей вуза.

**Объект исследования:** веб ресурс для обеспечения деятельности издательства ПГУ им. Т.Г.Шевченко.

**Предмет исследования:** разработка веб ресурса для обеспечения деятельности издательства ПГУ им. Т.Г.Шевченко.

**Практическая значимость** исследования заключается в создании веб ресурса, который позволит сократить время издания печатного материала, путем налаживания связей между авторами, работниками научной части, рецензентами и самим издательством.

**Цель исследования** спроектировать и разработать веб ресурс для обеспечения деятельности издательства ПГУ им. Т.Г.Шевченко.

Для реализации поставленной цели необходимо решить следующие задачи:

1. Изучить научную литературу по предметной области;
2. Изучить положения издательства ПГУ им. Т.Г.Шевченко;
3. Провести анализ существующих издательств;
4. Разработать диаграмму процессов, диаграмму состояний и карту веб ресурса;
5. Разработать веб ресурс на основе уже имеющихся сведений;

6. Провести тестирование веб ресурса;
7. Рассчитать экономическую эффективность веб ресурса.

В процессе исследования использовались такие общенаучные **методы исследования**, как системный подход, изучение и обобщение отечественной и зарубежной практики, моделирование, экономический и статистический анализ.

Авторы научных произведений, в которых рассматривалась проблема исследования: В.И. Казакова, Е.А. Зайцева и другие.

Структура выпускной квалификационной работы включает в себя введение, 3 главы, заключение, список использованной литературы, приложения.

# **ГЛАВА 1. Теоретические исследования по издательской деятельности**

## **1.1 Издательство, типография, основные понятия**

Издательство — предприятие, осуществляющее подготовку, выпуск и реализацию печатных, электронных и комбинированных изданий. В подготовку изданий входит:

- редактирование и корректура авторского текста;
- художественное и техническое редактирование, т. е. разработка оформления издания;
- составление издательской спецификации (технических данных об издании).

Типография — полиграфическое предприятие, выпускающее печатную продукцию и имеющее в своем составе производственные цеха или подразделения (допечатное, формное, печатное, брошюровочно-переплетное и др.), а также вспомогательные службы (ремонтно-механические, складские, лаборатории и т. п.).

Выпускаемая печатная продукция очень разнообразна по своему виду, конкретному назначению, срокам издания, техническому исполнению. В зависимости от потребительского назначения ее разделяют на следующие группы:

1. Периодические издания — газеты, журналы, регулярно выходящие сборники и т. д.
2. Книги и брошюры.
3. Листовые одно- и многокрасочные изобразительные издания — плакаты, портреты, репродукции произведений живописи, наглядные пособия, художественные открытки и др.
4. Упаковочную продукцию — этикетки, обертки и пр.
5. Деловую продукцию — бланки, техническую документацию на товары, оборудование, книги конторского учета, рекламные издания и т. д. 12

6. Специальную продукцию — денежные знаки, акции, бланки документов (паспорта, различные удостоверения), проездные билеты и т. д.
7. Представление издания или его элементов в цифровом виде (обработанного текста, обработанных иллюстраций, сверстанных полос, спуска полос). [27]

Издательство и типография при университете закрывают потребность в большом количестве печатной продукции, что позволяет более эффективно вести образовательный процесс и научную деятельность.

## **1.2 Анализ существующих веб ресурсов для типографии и издательской деятельности**

Так же необходимо провести анализ существующих веб ресурсов для типографии и издательской деятельности Приложение Н.

Каждая характеристика оценивается по баллам от 0-2, где 0 – полное несоблюдение критерия, 1 – частичное соблюдение, 2 – абсолютное соблюдение.

В результате сравнительного анализа мы выявили что максимальное соответствие характеристикам есть у «Tirar» типографии, несмотря на ее простоту. В то время как «Printstore Group» набрал минимальный результат из всей выборки. Несмотря на простоту «Tirar» типографии.

Сравнение по организационно-экономическим показателям предполагает использование бинарной системы оценки (да, нет) и указание абсолютных значений (для посещаемости ресурса) Приложении П.

В результате сравнения было выявлено что самый качественный ресурс имеет «Перо», так как есть показы за месяц и минимальное время загрузки.

Сравнительный анализ по удовлетворению требований, которые необходимы университету ПГУ им. Т.Г. Шевченко предполагает использование бинарной системы оценки (да, нет) и указание абсолютных значений (для посещаемости ресурса) таблица 1.

**Таблица 1 -Сравнительный анализ по удовлетворению требований**

| Требование                                   | Тіpar<br>типографія -<br><a href="https://tipar.ru/">https://tipar.ru/</a> | Printstore Group -<br><a href="https://printstore.com.ua/">https://printstore.com.ua/</a> | Пепо -<br><a href="https://pero-print.ru/">https://pero-print.ru/</a> | Artpoligraf -<br><a href="https://artpoligraf.md/">https://artpoligraf.md/</a> |
|--|--|---|---|--|
| Расчет стоимости печатного изделия           | нет  | да  | Да, но результат отправляется на почту                                | да   |
| Подача документов онлайн                     | нет  | нет   | нет   | да   |
| Публикация на elibrary                       | нет  | нет   | нет   | Нет  |
| Возможность публиковать в РИНЦ               | нет  | нет   | нет   | Нет  |
| Личный кабинет пользователя                  | нет  | да  | нет   | Да   |
| Подготовка печатных материалов перед печатью | Разработка макета  | нет   | да  | Нет  |

По итогам сравнительного анализа по удовлетворениям требований было выявлено что нет подходящего решений, поэтому решение будет разрабатываться самостоятельно.

### **1.3 Проект разрабатываемого программно-технического решения**

Проект разрабатываемого программно-технического решения всегда начинается с постановки проблемы, которое создаваемый проект будет решать.

Основная проблема, которая на данный момент есть в издательстве это достаточно долгий процесс обработки печатного издания из-за общения с клиентами по средствам почты.

Основные требования к разрабатываемому программному решению:

1. Веб ресурс должен обеспечивать прозрачность при расчете стоимости печатного издания;

2. Решение должно уменьшить время на обработку печатного издания;
3. Решение должно иметь удаленный доступ;
4. Веб ресурс должен иметь понятный, удобный и простой интерфейс;
5. Веб ресурс должен иметь регистрацию;
6. Решение должно быть адаптировано под разные роли пользователей.

Выделены 4 основные роли в веб ресурсе, а именно: пользователь, работник научной части, рецензент и работник издательства. Основные их возможности указаны на рисунке 1.



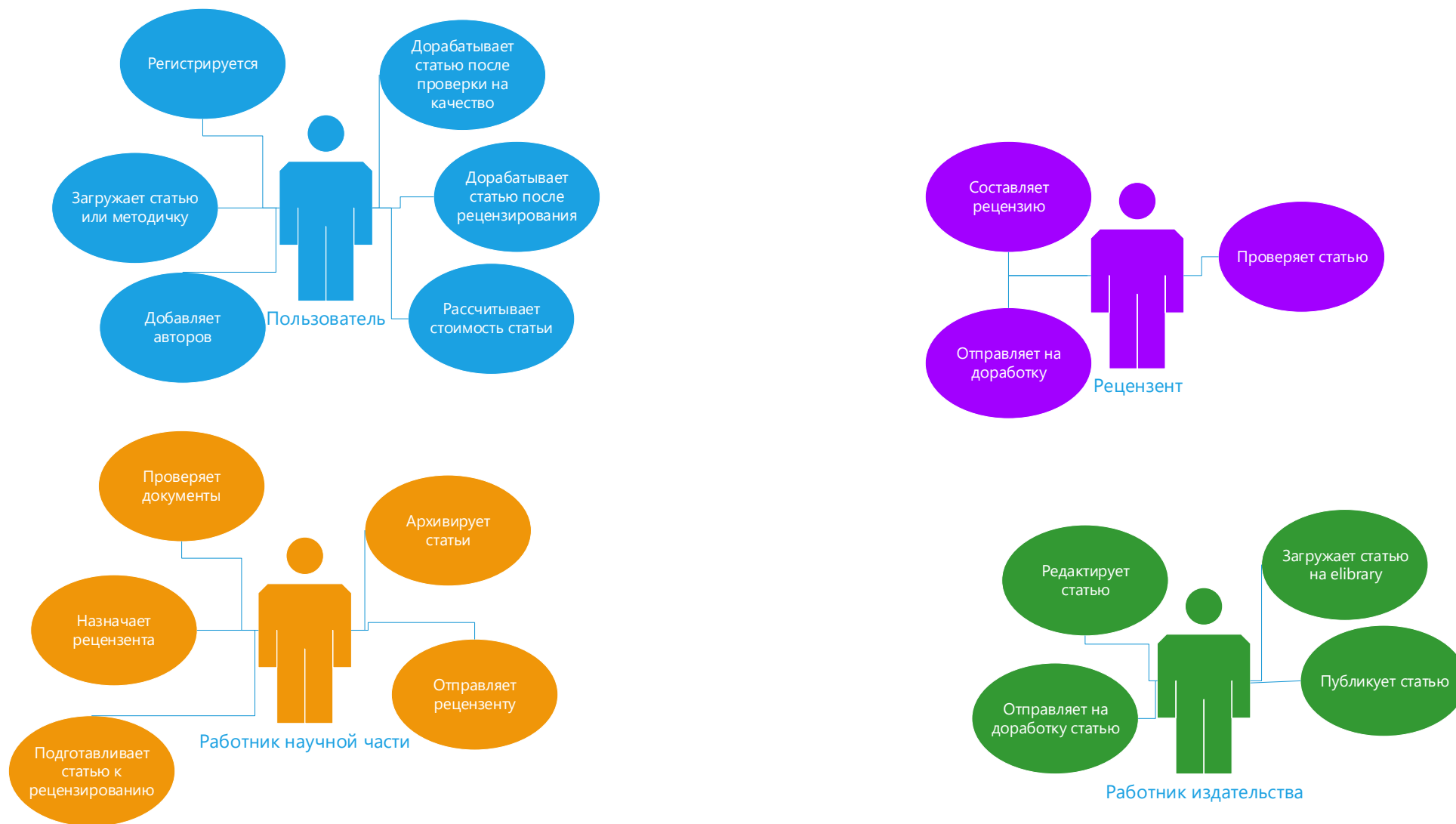


Рисунок 1 - Диаграмма состояний ролей



Рассмотрим на примере публикации статьи. После загрузки статья попадает к научной части, где просматривается корректный ли был отправлен файл, далее если все правильно статья подготавливается к отправки рецензенту и отправляется.

Рецензент видит статью без авторов, которые ее написали, для соблюдения слепого рецензирования, составляет рецензию и выносит свой вердикт. Так рецензент может отправить статью на публикацию, доработку и не рекомендовать к публикации. После того как статья проверена и направлена на публикацию, она отправляется в издательство, там тоже ее проверяют и отправляют либо на доработку, либо сразу на печать.

После того как статья всеми одобрена она загружается на eLibrary.Ru — российская научная электронная библиотека, интегрированная с Российским индексом научного цитирования. И в самом конце архивируется.

Вывод: в ходе анализа было выявлено что подходящего решения нет, поэтому необходимо создать собственное, которое уменьшит время обработки печатного издания, что в свою очередь позволит повысить количество изданий что в конечном итоге приведет к повешению репутации университета ПГУ им. Т.Г. Шевченко.

## **ГЛАВА 2. Проектирование веб ресурса для издательства ПГУ им. Т.Г. Шевченко**

### **2.1 Краткая характеристика издательства ПГУ им. Т.Г. Шевченко**

Издательство ведет свою деятельность начиная с октября 1993 года. Учитывая относительную отдаленность от российского книжного рынка и невозможность регулярных массовых поставок книжных новинок, деятельность Издательства Приднестровского университета, направленная на удовлетворение растущих потребностей общества республики в русскоязычной учебной и научной литературе, является крайне необходимой.

Важное место в работе издательства занимает подготовка и выпуск научно-методического журнала «Вестник Приднестровского университета». Журнал сегодня издается в трех сериях (гуманитарные, медико-биологические и химические, физико-математические и технические науки) три раза в год. В нем находят отражение наиболее значимые научные достижения профессорско-преподавательского коллектива вуза.

Издательство Приднестровского университета использует в своем производстве цифровые технологии оперативной полиграфии, уникальность которых состоит в возможности комплексного решения задач высшей школы по обеспечению научной, учебной и другой литературой всех структурных подразделений вуза. Высокопроизводительные цифровые комплексы оперативной печати обеспечивают:

- высокое полиграфическое качество выпускаемой продукции;
- масштабируемость тиражируемых материалов;
- централизованность управления полиграфическим производством;
- модульность технологического процесса издания литературы.

В структуру Издательства входят: редакционно-издательский и производственный отделы.

Директор Издательства:

- планирует работу Издательства, обеспечивает выполнение основных задач и функций Издательства;
- назначается и освобождается от занимаемой должности приказом ректора университета в установленном порядке.

Начальник производственного отдела направляет и контролирует деятельность сотрудников производственного отдела по изданию работ, внедряет новую технику в издательский процесс, отвечает за ее сохранность.

Начальник редакционно-издательского отдела контролирует и направляет творческий процесс в отделе по подготовке макетов изданий. Редактирует наиболее значимые работы. Делает контрольную вычитку макетов, отредактированных молодыми сотрудниками.

Штатное расписание Издательства формируется в соответствии с плановым объемом, характером выполняемых работ, установленными нормативами и утверждается ректором.

Издательство взаимодействует с институтами, факультетами, кафедрами, библиотекой, управлениями научной деятельности и академической политики, центральной бухгалтерией, финансовым управлением и другими подразделениями университета, сторонними организациями и учреждениями.

## **2.2 Анализ имеющейся системы взаимодействия в издательстве и необходимость ее совершенствования**

На данный момент издательство не имеет самостоятельного веб ресурса, и основная информация расположена на сайте университета ПГУ им. Т.Г. Шевченко. Связь с клиентами ведется по средствам e-mail почты.

Так же расчет стоимости печатного издания ведется в таблице Excel рисунок 3.

|    | A  | B            | C       | D        | E                   | F             | G         | H    | I | J | K | L | M | N | O | P | Q | R |
|----|--|--------------|---------|----------|---------------------|---------------|-----------|------|---|---|---|---|---|---|---|---|---|---|
| 1  | статья   | норма, руб   | участие | факт     | тираж               | 1             |           |      |   |   |   |   |   |   |   |   |   |   |
| 2  | редактирование                                 | 58,7         | 0       | 0        | объем               | 10            |           |      |   |   |   |   |   |   |   |   |   |   |
| 3  | первая корректура                              | 26,96        | 0       | 0        |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 4  | ревизия  | 15,48        | 0       | 0        |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 5  | набор  | 22,61        | 0       | 0        |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 6  | макетирование                                  | 16,61        | 0       | 0        |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 7  | тиражирование                                  | 39,18        | 1       | 3 918    |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 8  | оплата труда                                   | 179,54       |         | 3 918    |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 9  | отчисления в бюджет 25%                        | 44,885       |         | 0,9795   |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 10 | оплата товаров и услуг                         | 27,32895485  |         | 524,5539 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 11 | бумага, руб                                    | 0,024003     |         | 8,160795 | 0,24003 кг          |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 12 | мастер-пленка                                  | 0,018518519  |         | 388,3326 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 13 | краска   | 0,00         |         | 21,93883 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 14 | клей   | 0,00         |         | 0,66585  |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 15 | скобы  | 0,10         |         | 0,26     |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 16 | прочие расходные (включая допечатные процессы) | 0,25         |         | 104,6081 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 17 | расходные материалы                            | 0,397954852  |         | 523,9662 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 18 | хозрасходы (комм. усл.) (15% ФОТ)              | 26,931       |         | 0,5877   |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 19 | приобретение ОФ (включая износ)                | 0,12         |         | 1,2      |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 20 | итого затрат                                   | 224,9429549  |         | 530,6514 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 21 | цена экземпляра                                | 224,9429549  |         | 530,6514 |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 22 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 23 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 24 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 25 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 26 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 27 |  |              |         |          |                     |               |           |      |   |   |   |   |   |   |   |   |   |   |
| 28 | ЦВ печать                                      |              | норма   | факт     | 20 тираж            | бумага, А3 кг | стоимость |      |   |   |   |   |   |   |   |   |   |   |
|    |  | расходные ме |         | 480      | 6 кол-во страниц А4 | 0,9936        | 29,808    | цена |   |   |   |   |   |   |   |   |   |   |

Рисунок 3 - Расчет стоимости печатного издания в таблице Excel

Основной недостаток организации работы таким образом заключается в том, что чтобы узнать стоимость печати необходимо обратиться напрямую в издательство.

В издательстве используются следующие понятия при печати.

### Авторский лист

Это единица объема авторского оригинала, т. е. материала, созданного автором (текста, таблиц, иллюстраций). Она равна:

- прозаический текст: 40 000 знаков текста (считая за знак и каждый пробел между словами, и каждый знак препинания, и цифру и т. п., и неполные строки за полные);
- стихотворный текст: 700 строк;
- иллюстративный материал: 3 000 см<sup>2</sup> площади изображения на полосе книги.

### Учетно-издательский (физический печатный) лист

Это единица объема всего материала книги (кроме рекламных текстов). Она исчисляется так же, как авторский лист, но помимо материала, созданного автором, сюда входит еще и то, что внесено в книгу издательством: выходные сведения на титульном листе, обложке (переплете), корешке, выпускные данные, вступительная статья, редакционное предисловие и послесловие, повторяющиеся головки таблиц, колонцифры и т. п.

Учетно-издательский лист – это единица натурального объема изданий (бумаги с печатью). Физически – это площадь поверхности (стороны)

бумажного листа, воспринявшего краску с печатной формы того же размера. Такой печатный лист также называется физическим печатным листом.

Вузовские издания обычно печатаются в формате 60×90/16. Следовательно, учетно-издательский лист в таких изданиях составляет 16 страниц.

Пример подсчета объема книги в учетно-издательских листах. В книге, изданной форматом 60×90/16, 160 страниц. В каждом печатном листе 16 страниц. Объем книги в учетно-издательских (физических печатных) листах составит:  $160 : 16 = 10$ .

### **Условный печатный лист**

Данный объемный показатель используется при издании книг, журналов на бумаге отличных от 60×90 форматов. Чтобы можно было сопоставить печатный объем разноформатных книг, был введен общий измеритель – условный печатный лист форматом 60×90 см. К нему приводятся печатные листы всех других форматов. Отсюда другое его название – приведенный печатный лист или условный печатный лист.

Пример перевода учетно-издательских листов в условные печатные. Журнал форматом 70×100/16 содержит 160 страниц, т. е. 10 учетно-издательских листов. Чтобы узнать, чему равен объем этой книги в условных печатных листах, нужно использовать коэффициент перевода. Так, один учетно-издательский лист формата 70×100 соответствует одному условно-печатному листу, умноженному на коэффициент перевода 1,29. В нашем примере  $160 : 16 \times 1,29 = 12,9$ .

Таким образом, объемные показатели 160-страничного журнала форматом 70×100/16 будут следующими: 10 учетно-издательских листов или 12,9 условных печатных листов.

Зная коэффициенты перевода учетно-издательских листов всех стандартных форматов, отличных от формата 60×90, можно подсчитать объем любой книги в условных печатных листах, умножив ее объем в физических печатных листах на переводной коэффициент.

Так несмотря на то, что есть формула расчета стоимости в самом издательстве можно будет узнать более точную цену.

## **2.3 Структура и содержание веб ресурса**

### **2.3.1 Логическая структура ресурса**

Для того чтобы создать веб ресурс необходимо составить его логическую структуру, рисунок 4. Можно выделить 4 основные блока разработки, а именно: вход, кабинет администратора, кабинет пользователя и страница рецензента.

Все начинается с главной страницы, где пользователь выбирает что он хочет сделать, зайти или посчитать сколько стоит печатное издание.

Основные параметры для расчетов — это тираж, объем и услуги, которые может предоставить издательство. Так можно получить себестоимость печатного издания или за весть объем работ.



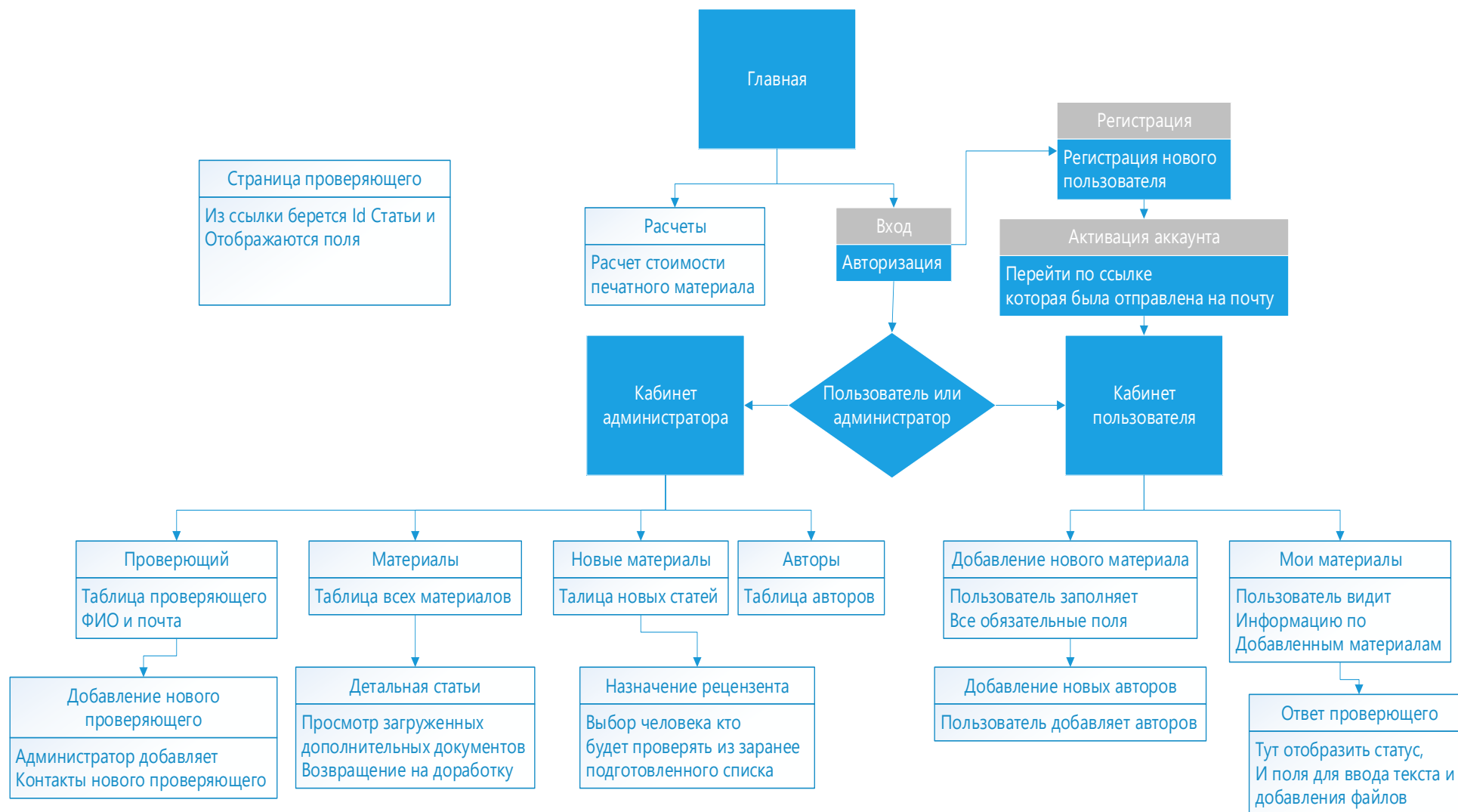


Рисунок 4 - Карта сайта

Для входа в систему нужна будет только почта и пароль пользователя, а для того, чтобы зарегистрироваться нужно будет ввести ФИО и пароль два раза. После на почту придет письмо, в котором будет ссылка на активацию аккаунта, после которого пользователь сможет загружать свои материалы.

Для того чтобы загрузить свои материалы дополнительно потребуется внести в систему небольшую информацию об авторе, в дальнейшем это поможет осуществлять по ним фильтрацию. Кроме того, у пользователя будет раздел, где будут храниться его материалы с детальной информацией по каждому ресурсу.

Основной блок — это блок администратора, где работник научной части отправляет материал на проверку рецензенту, и работник издательства, который тоже проверяет и направляет на доработку или печать.

И последний блок — это страница рецензента, на почту рецензента приходит письмо о том, что был получен новый материал и его необходимо проверить. Так на пример если мы говорим о проверки статьи, то на этой странице будет сама статья, но без авторов, для соблюдения правила слепого рецензирования, основная информация по статье и поля для добавления комментария и рецензии.

### 2.3.2 Параметры и внешний вид ресурса

Для создания веб ресурса были выбраны постельные бежевые тона, а именно рисунок 5.

| Цвет №1   | Цвет №2   | Цвет №3   | Цвет №4  | Цвет №5   |
|---|---|---|--|---|
|  |  |  |  |  |
| #bb875e   | #ddd4c2   | #dedfd6   | #c1ad98  | #d1c6b6   |
| rgb(187,135,94)   | rgb(221,212,194)  | rgb(222,223,214)  | rgb(193,173,152)   | rgb(209,198,182)  |

Рисунок 5 - Основная палитра цветов

Так, например главная страница выглядит следующим образом, рисунок 6.



Рисунок 6 - Главная страница веб ресурса

Следующая страница отвечает за вход, рисунок 7.

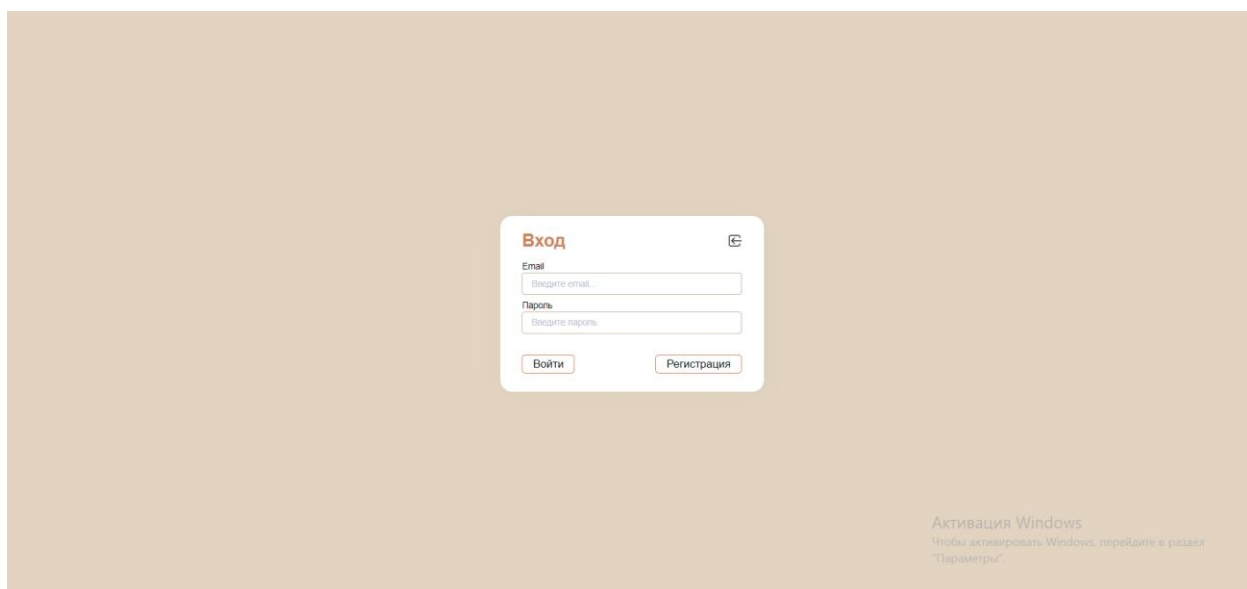


Рисунок 7 - Вход в веб ресурс

Издательство ПГУ  
им. Т.Г. Шевченко

Ярослава Сергеевна [Выход](#)

Добавить автора

Добавить материал

Мои материалы

**Добавить автора**

Имя

Фамилия

Отчество

Email

Контакты

☒ Нет, не преподаватель

Место работы

Должность

[Сохранить автора](#) [Отмена](#)

Рисунок 8 – Добавление автора в веб ресурс

### 2.3.3 Основные продукты для разработки

Для разработки было выбран фреймворк Vue 3, язык разработки на клиентской стороне пользовательского интерфейса TypeScript, язык разработки на программно-аппаратной части сервиса, отвечающей за функционирование его внутренней части используется SQL и C#, а именно Swagger, EntityFramework, ASP.NET Core.

Программы необходимы для работы: SQL Server, Microsoft SQL Server Management Studio, Visual Studio 2022, Visual Studio Code.

Фреймворк — это программный продукт, который упрощает создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений.

Vue (произносится /vju:/, примерно, как view) — это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения.

Основные преимущества Vue:

- Доступность - создан на основе стандартных HTML, CSS и JavaScript с интуитивно понятным API и документацией мирового класса.
- Исполнительность - по-настоящему реактивная, оптимизированная компилятором система рендеринга, которая редко требует ручной оптимизации.
- Универсальность - богатая, постепенно адаптируемая экосистема, которая масштабируется между библиотекой и полнофункциональной платформой.

TypeScript — это строго типизированный язык программирования, основанный на JavaScript, предоставляющий лучшие инструменты в любом масштабе.

Основные моменты в TypeScript:

- TypeScript добавляет в JavaScript дополнительный синтаксис для обеспечения более тесной интеграции с вашим редактором. Позволяет выявлять ошибки на ранней стадии.
- Код TypeScript преобразуется в JavaScript, который запускается везде, где запускается JavaScript: в браузере, на Node.js или Deno и в ваших приложениях.
- TypeScript понимает JavaScript и использует вывод типов, чтобы предоставить вам отличные инструменты без дополнительного кода[31].

SQL — это структурированный язык запросов, созданный для того, чтобы получать из базы данных необходимую информацию. Если описать схему работы SQL простыми словами, то специалист формирует запрос и направляет его в базу. Та в свою очередь обрабатывает эту информацию, «понимает», что именно нужно специалисту, и отправляет ответ.

Данные хранятся в виде таблиц, они структурированы и разложены по строкам и столбцам, чтобы ими легче было оперировать. Такой способ хранения информации называют реляционными базами данных (от англ.

relation — «отношения»). Название указывает на то, что объекты в такой базе связаны определенными отношениями.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 2019, которая вышла в 2019 году, и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

SQL Server характеризуется такими особенностями как:

- Производительность. SQL Server работает очень быстро.
- Надежность и безопасность. SQL Server предоставляет шифрование данных.
- Простота. С данной СУБД относительно легко работать и вести администрирование.

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, Базы данных SQL Azure, Управляемого экземпляра SQL Azure, SQL Server на виртуальной машине Azure и Azure Synapse Analytics. Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

C# (произносится как «See Sharp») — это современный, объектно-ориентированный и типобезопасный язык программирования. C# позволяет

разработчикам создавать множество типов безопасных и надежных приложений, работающих в .NET. C# берет свое начало в семействе языков C и будет сразу знаком программистам на C, C++, Java и JavaScript.

Несколько функций C# помогают создавать надежные и долговечные приложения. Сборка мусора автоматически освобождает память, занятую недоступными неиспользуемыми объектами. Типы, допускающие значение NULL, защищают от переменных, которые не относятся к выделенным объектам. Обработка исключений обеспечивает структурированный и расширяемый подход к обнаружению и устранению ошибок. Лямбда-выражения поддерживают методы функционального программирования. Синтаксис Language Integrated Query (LINQ) создает общий шаблон для работы с данными из любого источника. Языковая поддержка асинхронных операций обеспечивает синтаксис для построения распределенных систем. C# имеет унифицированную систему типов. Все типы C#, включая примитивные типы, такие как `int` и `double`, наследуются от одного корневого `object`-типа. Все типы имеют набор общих операций. Значения любого типа можно хранить, транспортировать и оперировать согласованным образом. Кроме того, C# поддерживает как определяемые пользователем ссылочные типы, так и типы значений. C# допускает динамическое размещение объектов и встроенное хранение легковесных структур. C# поддерживает универсальные методы и типы, которые обеспечивают повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять настраиваемое поведение клиентского кода.

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства .NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц. Также код с использованием EF пишется гораздо быстрее.

Например, работая с базами данных напрямую, разработчик должен беспокоиться о подключении, подготовке SQL и параметров, отправке

запросов и транзакций. На Entity Framework всё это делается автоматически — программист же работает непосредственно с сущностями и только говорит EF, что нужно сохранить изменения.

Swagger — это набор инструментов, которые помогают описывать API. Благодаря ему пользователи и машины лучше понимают возможности REST API без доступа к коду. С помощью Swagger можно быстро создать документацию и отправить ее другим разработчикам или клиентам.

ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету.

ASP.NET Core позволяет выполнять следующие задачи:

- Создавать веб-приложения и службы, приложения Интернета вещей (IoT) и серверные части для мобильных приложений.
- Использовать избранные средства разработки в Windows, macOS и Linux.
- Выполнять развертывания в облаке или локальной среде.
- Запускать в .NET Core.

Интегрированная среда разработки (IDE) — это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

Вывод: в ходе работы выяснилось что для реализации всех задач и продуктов ресурса необходимо затратить большое количество времени.



## **ГЛАВА 3. Создание веб ресурса для издательства ПГУ им. Т.Г. Шевченко**

### **3.1 Разработка веб ресурса**

Первое с чего следует начать это с описания базы данных Приложение Б. Проще всего ее можно понять, двигаясь от таблицы с факультетами, к таблице с кафедрами. Таблица факультет имеет с таблицей кафедры связь один ко многим что значит есть одна кафедра и у нее может быть много факультетов.

Кафедры и факультеты нам нужны для описания автора, которого мы добавляем, если он является преподавателем в университете.

В таблице автора обозначены основные поля, которые нужны издательству для работы, а именно: ФИО, почта и место работы, контактные данные.

Следующая таблица с авторами и публикациями является связующей с таблицей самой публикации. Она помогает не нагружать базу данных и делать легкие и быстрые запросы.

Сама таблица публикации имеет свои поля, а так же связь с таблицами файлов и с таблицей рецензентов.

Так же используется таблица пользователей, которые после активации почты входят в свои учетные записи и добавляют публикации, добавляют авторов.

Таблица с mailToken необходима для активации аккаунта.

Так же рассмотрим явное указание ключей для работы связующей таблицы авторов и публикаций, Приложение А, в коде говорится что один автор может быть записан во многих публикациях и внешним ключом будет выступать AuthorId, в то время как у одной публикации может быть много авторов и внешним ключом будет выступать PublicationId, а так же оба эти поля обязательны.

Следующее что мы рассмотрим это работу контроллера, а именно задачу добавления публикации, Приложение В. Вначале мы указываем что это Post

запрос, после что данный запрос могут делать только авторизированные пользователи, а так же по какому пути будет передаваться запрос, в наше случае Publication/Add.

Далее в тела запроса мы берем модель которую нам пришла, вызываем задачу из сервиса по добавлению, приложение Г, туда передаем модель, там ее сопоставляем с нашей моделью, после запись добавляется в таблицу базы данных, база сохраняется и возвращается результат сохранения, успешно или была какая-то ошибка. Возвращаем мы результат в базовой модели, которая имеет следующий вид приложение Д.

Если говорить о том, как делается запросы на клиентской части, то все начинается с описание сервиса, а именно: путь обращения, модели запроса и ответа, указание метода для обращения Приложение Е. Данный сервис реализован для работы с публикацией.

В нем вызывается BaseRequestService, приложение Ж, это общий сервис для создания запроса, в нем вызывается метод sendXHR которым происходит отправка запроса приложение И.

Для начала строится путь, который будет в запросе, в нем указывается, где будут параметры, в пути или в теле.

<https://localhost:7212/Publication/Get> - данный запрос вернет все публикации что есть в базе.

<https://localhost:7212/Publication/Get?publicationId=1> – данный запрос вернет детальную информацию о публикации с id равной 1.

После указывается обработка запроса и заполняется тело запроса.

Так же стоит отметить, важность роутинга на клиентской части, он отвечает за все пути, по которым может перейти клиент, Приложение К. Данный роутинг предназначен администратору ресурса. Модель роута включает в себя имя, по нему мы в коде обращаемся к той или иной странице, путь, это тот путь, который будет виден клиенту, чаще всего он совпадает с именем, сама страница, которая должна отображаться. А также может содержать redirect – это то куда нужно отвести клиента дальше по пути, в meta

содержится дополнительная информация о странице и children если роут общий.

По мимо роутов необходимо сказать и о middleware – функция промежуточной обработки, Приложение Л. Данная функция позволяет допускать к определенным страницам только авторизованных пользователей и только в соответствии их роли.

В приложении М показан компонент, который создан с помощью vue-property-decorator. Данный компонент содержи template в котором хранится все что видит пользователь, script в котором выполняется код и style в котором используются стили для данного компонента.

С помощью vue-property-decorator мы можем использовать декораторы - они позволяют использовать определение Vue-компоненты в виде полноценного класса. Соответственно, появляется возможность определения всех свойств и методов Vue-компоненты в явном виде.

### **3.2 Тестирование веб ресурса**

Swagger UI визуализирует ресурсы OpenAPI и взаимодействие с ними без отображения логики реализации. Этот инструмент берет спецификацию и превращает ее в интерактивный проект, где можно выполнять разные запросы для тестирования API.

Плюсы Swagger UI:

- Нет зависимостей. Интерфейс доступен в любой среде разработки.
- Полная поддержка браузерами.
- Удобное взаимодействие для других разработчиков и клиентов. Можно проверить все операции, предоставляемые API, даже не имея специальных знаний.
- Простая навигация. Ресурсы и конечные точки представлены в виде аккуратно распределенного на категории списка.
- Можно настраивать стиль и пользовательский интерфейс так, как хочется.

В Swagger имеется описание API и общие модели.

Тестирование начинается прежде всего с авторизации пользователя, а именно в тело запроса вход почта и пароль, рисунок 9.

# Auth

POST

/Auth/Login

⬆️🔒

Parameters

Cancel

| Name                               | Description                                       |
|------------------------------------|---|
| email <small>* required</small>    | <input type="text" value="yaroslav2001@list.ru"/> |
| string<br>(query)                  |   |
| password <small>* required</small> | <input type="text" value="123"/>                  |
| string<br>(query)                  |   |

Execute

Clear

### Рисунок 9 - Запрос на вход в веб ресурс

| Responses  | Response content type  |
|--|--|
| <pre>curl -X 'POST' \ 'https://localhost:7212/Auth/Login?email=yaroslavas2001&amp;list.ru&amp;password=123' \ -H 'accept: application/json' \ -H 'Authorization: AAAABE85C-DCB2-4FCD-B4FC-75CF05AA985' \ -d ''</pre> | application/json   |
| Request URL  | <pre>https://localhost:7212/Auth/Login?email=yaroslavas2001&amp;list.ru&amp;password=123</pre>   |
| Server response  |  |
| Code   | Details  |
| 200  | <pre>{   "data": {     "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZWlkeS46bmVkbWZfcmw5YzY3cyByeDA1LzA1L2lkZW50aXRSL2NsYWltcy9uYmlljoiMSIsInR5cCI6IjE6MTY1InMwPjY1MSwiZXhwIjoxNjU2OTQyOQJjLjP.c3I0LjZ7dHcvZmh0LiIpSiZKIGllhcmwzbGF2MI19.1387FAIxenSTnkSm6C2SKSOKTX173JorMkHbhm6UJq",     "firstName": "Аполонская",     "sureName": "Сепремина",     "role": 0,     "id": 1   },   "isSuccess": true,   "errorMessage": null,   "stackTrace": null,   "errorCode": 0 }</pre> |

Рисунок 10 - Ответ на запрос входа в веб ресурс

Запрос успешно прошел, рисунок 10 и вернул код 200. Так же вернул информацию о пользователе, токен на пользование веб ресурсом и роль пользователя.

В сервисе входа доступны следующие методы, рисунок 11.

| Auth |                       | ^   |
|------|-----------------------|-----|
| POST | /Auth/Login           | ▼ 🔒 |
| POST | /Auth/Register        | ▼ 🔒 |
| POST | /Auth/ActivateAccount | ▼ 🔒 |

Рисунок 11 - Методы для входа в веб ресурс

Следующий сервис необходим для добавления автора в систему, рисунок 12.

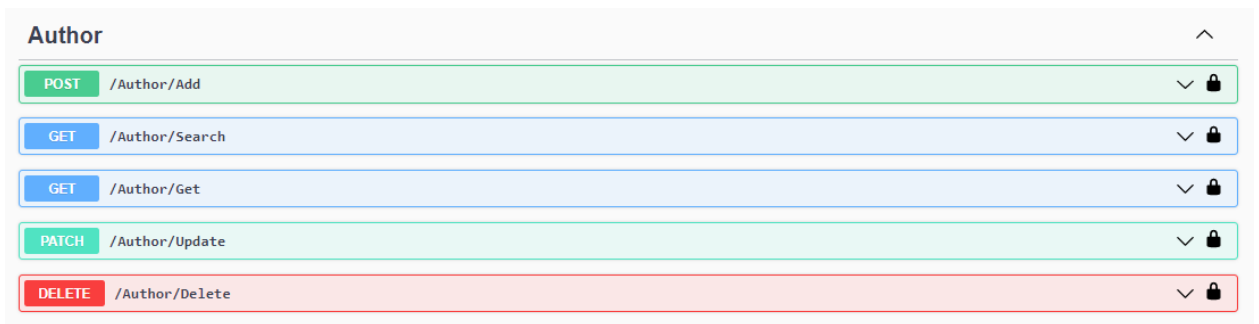


Рисунок 12 - Методы для работы с автором  
Для тестирования выберем метод добавления, рисунок 13.

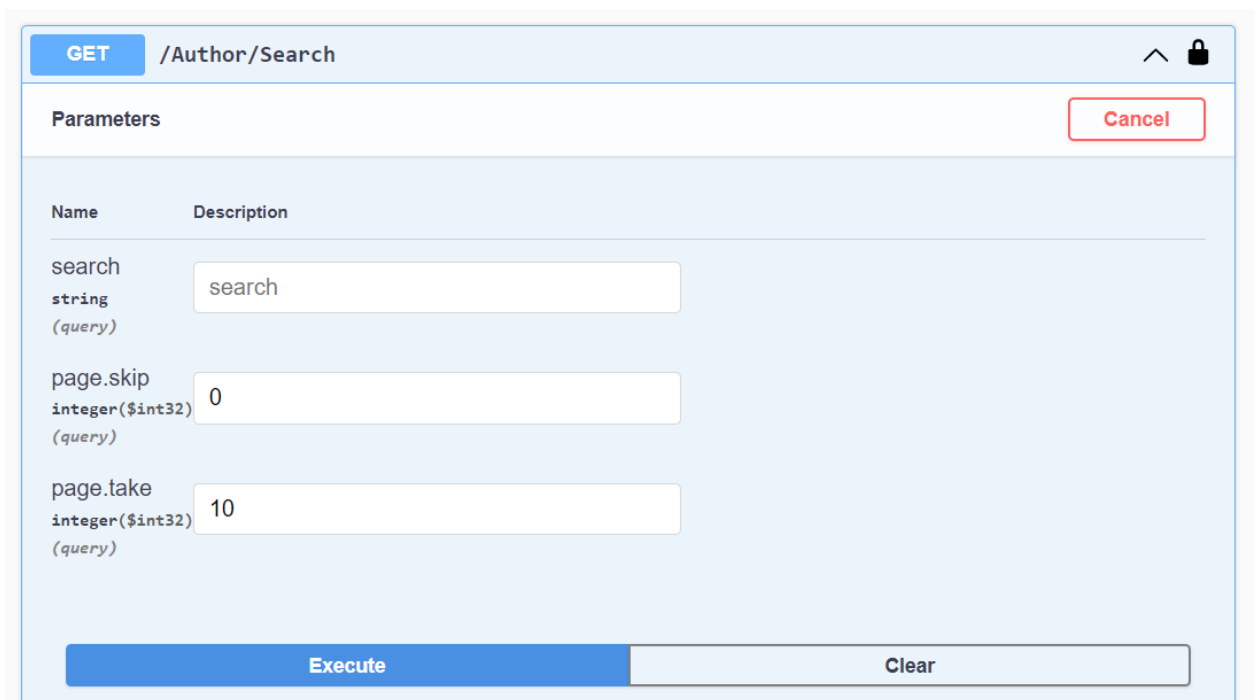


Рисунок 13- Модель запроса поиска по авторам  
В результате мы получаем рисунок 14.

Responses Response content type application/json

Curl

```
curl -X 'GET' \
  'https://localhost:7212/Author/Search?page.skip=0&page.take=10' \
  -H 'accept: application/json' \
  -H 'Authorization: AAA0EB5C-DCB2-4FCD-B4FC-75CFC054A985'
```

Request URL

```
https://localhost:7212/Author/Search?page.skip=0&page.take=10
```

Server response


| Code | Details  |
|------|--|
| 200  | <p>Response body</p> <pre>{   "data": {     "page": {       "skip": 0,       "take": 10     },     "count": 2,     "items": [       {         "id": 1,         "firstName": "Ирина",         "sureName": "Руслановна",         "secondName": "Партас"       },       {         "id": 2,         "firstName": "Егор",         "sureName": "Анатольевич",         "secondName": "Ставничий"       }     ]   },   "isSuccess": true,   "errorMessage": null,   "stackTrace": null,   "errorCode": 0 }</pre> <p> Download</p> |

Рисунок 14 - Ответ на запрос поиска всех авторов

Сервисы для работы с факультетами и кафедрами выглядят следующим образом, рисунок 15.

| Departmane |                    |                                  | ^   |
|------------|--------------------|----------------------------------|-----|
| POST       | /Department/Add    | Добавить кафедру                 | ⌵ 🔒 |
| GET        | /Department/GetAll | Получить список всех кафедр      | ⌵ 🔒 |
| PATCH      | /Department/Rename | Переименовать Кафедру            | ⌵ 🔒 |
| DELETE     | /Department/Delete | Удалить кафедру                  | ⌵ 🔒 |
| Faculty    |                    |                                  | ^   |
| POST       | /Faculty/Add       | Добавить факультет               | ⌵ 🔒 |
| GET        | /Faculty/GetAll    | Получить список всех факультетов | ⌵ 🔒 |
| GET        | /Faculty/Get       | Получить факультет               | ⌵ 🔒 |
| PATCH      | /Faculty/Rename    | Переименовать факультет          | ⌵ 🔒 |
| DELETE     | /Faculty/Delete    | Удалить факультет                | ⌵ 🔒 |

## Рисунок 15 - Сервисы для работы с кафедрами и факультетами

В основном данные сервисы нужны только для детальной информации автора если он работает в университете, но так как кафедр достаточно много было принято решение сделать сервис, а не список кафедр.

| File              |                                |  | ^   |
|-------------------|--------------------------------|--|-----|
| PUT               | /File/UploadFileForPublication | Загрузить файл и прикрепить его к публикации | ✓ 🔒 |
| GET               | /File/Get                      | Получить файлы публикации                    | ✓ 🔒 |
| Publication       |                                |  | ^   |
| POST              | /Publication/Add               | Добавить публикацию                          | ✓ 🔒 |
| GET               | /Publication/Get               | Получить все публикации                      | ✓ 🔒 |
| PATCH             | /Publication/SetStatus         | Установить статус публикации                 | ✓ 🔒 |
| PATCH             | /Publication/Update            | Обновить публикацию                          | ✓ 🔒 |
| PublicationAuthor |                                |  | ^   |
| POST              | /PublicationAuthor/Add         | Установить автора публикации                 | ✓ 🔒 |
| GET               | /PublicationAuthor/Get         | Получить авторов публикации                  | ✓ 🔒 |
| DELETE            | /PublicationAuthor/Remove      | Удалить автора публикации                    | ✓ 🔒 |

## Рисунок 16 - Сервисы для работы с публикацией

Так же необходимо протестировать сервис по работе с публикацией, а именно смена статуса рисунок 17.





| Review   |                           |   | ^   |
|----------|---------------------------|---|-----|
| POST     | /Review/Add               | Добавить ревью                              | ✓ 🔒 |
| GET      | /Review/Get               | Получить все ревью публикации               | ✓ 🔒 |
| GET      | /Review/Delete            | Удалить ревью                               | ✓ 🔒 |
| Reviewer |                           |   | ^   |
| POST     | /Reviewer/Add             | Добавить ревьюера                           | ✓ 🔒 |
| GET      | /Reviewer/Get             | Получить ревьюеров                          | ✓ 🔒 |
| GET      | /Reviewer/GetPublications | Получить идентификаторы публикаций ревьюера | ✓ 🔒 |

Рисунок 19 - Сервисы для рецензирования статьи

В итоге все методы при тестировании отрабатывают, особое внимание стоит уделить авторизации, так как если не авторизоваться в данном веб ресурсе можно получить коды с предупреждением что пользователь не зарегистрирован.

Так с помощью Swagger обычный человек может тестировать API что в очень хорошо для дальнейшей поддержки веб ресурса.

### 3.3 Расчет экономической эффективности веб ресурса

Методика расчета эффективности реализации проекта состоит из четырех этапов:

1. Оценка и анализ общих инвестиционных издержек. Предполагает расчет потребностей в основном и оборотном капитале, распределение потребностей в финансировании по стадиям инвестиционного цикла (проектирование, строительство, монтаж, пуско-наладка, выход на проектную мощность, работа на полной мощности);
2. Оценка и анализ текущих затрат. Сюда относится составление сметы затрат на производство продукции (работ, услуг), определение и анализ себестоимости отдельных видов продукции (работ, услуг);
3. Расчет и анализ показателей коммерческой эффективности проекта;
4. Определение показателей бюджетной эффективности.

Вначале необходимо определиться с графиком выполнения работ по разработке веб ресурса. Данные представлены в таблице 2. Для расчёта затраченного времени используются часы.

Таблица 2 - График выполнения работ

| Наименование работ                    | Длительность работы |            |
|---------------------------------------|---------------------|------------|
|                                       | в днях              | в часах    |
| Разработка технического задания       | 15                  | 60         |
| Планирование web-приложения           | 25                  | 75         |
| Рабочее проектирование web-приложения | 70                  | 280        |
| Отладка и тестирование web-сайта      | 10                  | 40         |
| Обобщение и оценка результатов        | 10                  | 40         |
|                                       | <b>130</b>          | <b>495</b> |

По результатам графика выполнения работ на разработку приложения было затрачено 495 человеко-часов.

Необходимо рассчитать стоимости разработки приложения, для этого необходимо учесть следующие виды расходов:

- 1) Основная заработная плата и отчисления в фонды;
- 2) Затраты на машинное время;
- 3) Стоимость материалов и покупных изделий;
- 4) Накладные расходы.

При разработке веб ресурса общее время разработки составило 5 месяцев. Из них машинное время составило 4,5 месяцев. Зная количество затраченного времени и принимая во внимание тот факт, что стипендия студента в месяц равна 242 рубля, произведен расчет фонда заработной платы, представлен в таблице 4.

Фонд оплаты труда за время работы над программным продуктом рассчитывается по формуле 4:

$$З_{\text{ФОТР}} = \sum O_{Pj} * T_{\text{РПР}j} * (1 + k_d) * (1 + k_y); \quad (4)$$

где  $O_{Pj}$ -оклад работника;

$T$ -количество периодов начисления;

$k_d$  - коэффициент дополнительной зарплаты;

$k_y$ - коэффициент районной зарплаты.

Так как работа выполнялась студентом, то  $k_d$  и  $k_y$  отсутствуют.

Таким образом,

$$З_{\text{ФОТР}} = 242 * 5 = 1210 \text{ руб.}$$

В соответствие с законом Приднестровской Молдавской Республики «О едином социальном аналоге и обязательном страховом взносе» (текущая редакция по состоянию на 15 января 2019 года) ставка Единого социального налога в Единый государственный фонд социального страхования составляют 25% от выплат, начисленных в пользу каждого работника, в том числе:

- на цели пенсионного страхования (обеспечения) -20,7%;
- на цели социального страхования работающих граждан – 3,0%;
- на цели страхования от безработицы – 0,3%;
- на улучшение оснащённости учреждений здравоохранения медицинским оборудованием и приобретение специального медицинского автотранспорта – 1%.

Сумма начислений на заработную плату составляет:

$$З_{\text{овф}} = 1210 * 0,25 = 302,5 \text{ руб.}$$

Для расчёта затрат, связанных с использованием вычислительной и оргтехники, необходимо знать машинное время работы над программным компьютером 475 часов, электропотребление компьютера за один час работы 0,12кВт, количество часов электропотребления 475. Расчёт затрат, связанных с использованием вычислительной и оргтехники, рассчитывается по формуле 5:

$$З_{\text{ЭВМ}} = P * K * C_{1\text{кВт/ч}} ; \quad (5)$$

где P-мощность энергопотребителей для АРМ программиста;

K – количество часов энергопотребления;

$C_{1\text{кВт/ч}}$ -стоимость потребления одного кВт/ч.

На работу ЭВМ за весь период расходуется  $475 * 1,2 = 570$  кВт/ч электроэнергии. В соответствие с постановлением Правительства ПМР от

31.05.2018 г. № 174 за объем электроэнергии, потреблённой до 150 Квт/ч, установлен тариф 0,54 руб за каждый кВт/ч.

Таким образом,

$$З_{\text{эвм}} = 1,2 * 475 * 0,54 = 307,8 \text{ руб}$$

Затраты на программные продукты рассчитываются по формуле 6:

$$З_{\text{спп}} = \sum C_p * k; \quad (6)$$

где  $C_p$ - цена единицы программного продукта;

$k$  - количество программных продуктов.

Расчёт затрат на программные продукты представлен в таблице 3.

Таблица 3 - Затраты на программные продукты

| Наименование       | Единица измерения | Количество | Цена за единицу, руб. | Стоимость, руб. |
|--------------------|-------------------|------------|-----------------------|-----------------|
| Visual Studio      | шт                | 1          | 3700                  | 3700            |
| My SQL             | шт                | 1          | 7200                  | 7200            |
| MS Office          | шт                | 1          | 1948                  | 1948            |
| Rider              | шт                | 1          | 1000                  | 1000            |
| Visual Studio Code | шт                | 1          | 0                     | 0               |
| Итого              | X                 | X          | X                     | 13848           |

Затраты на хозяйственно-организационные нужды рассчитываются по формуле 7:

$$З_{\text{хон}} = \sum C_r * K_r; \quad (7)$$

где  $C_r$ - цена единицы хозяйственно-организационной единицы;

$K_r$  – количество затраченных хозяйственно-организационных нужд.

Расчёт затрат на хозяйственно-организационные нужды представлен в таблице 4.

Таблица 4 – Затраты на хозяйственно-организационные нужды

| Наименование                         | Единица измерения | Количество | Цена за единицу, руб. | Стоимость, руб. |
|--------------------------------------|-------------------|------------|-----------------------|-----------------|
| Пакет Internet<br>Оптики " Optimum " | шт                | 5          | 180                   | 900             |
| Канцтовары<br>(тетрадь + ручка)      | шт                | 1          | 12                    | 12              |
| Бумага формата А4                    | упаковка          | 1          | 55                    | 55              |
| Компакт-диск с<br>программами        | шт                | 1          | 10                    | 10              |
| Итого                                | X                 | X          | X                     | 977             |

Также необходимо учесть накладные расходы, которые рассчитываются по формуле 8:

$$З_{\text{н}} = \Phi_{\text{отр}} * k_{\text{нр}}; \quad (8)$$

$k_{\text{нр}}$  составляют 30% от фонда заработной платы. Таким образом,

$$З_{\text{н}} = 1210 * 0,3 = 363 \text{ руб.}$$

В таблице 5 представлены все затраты на разработку программного продукта.

Таблица 5 – Итоговая смета затрат

| Наименование статей расходов  | Сумма, руб. |
|---|-------------|
| Общий фонд оплаты труда разработчика                                    | 1210        |
| Начисления на заработную плату  | 302,5       |
| Затраты, связанные с эксплуатацией техники                              | 307,8       |
| Затраты на специальные программные продукты, необходимые для разработки | 13848       |
| Затраты на хозяйственно-операционные нужды                              | 977         |
| Накладные расходы   | 363         |
| Итого   | 17008,3     |

Цена веб ресурса(Ц) определяется итоговыми затратами и прибылью, которая, в свою очередь, составляет 30% от фонда заработной платы. Таким образом, цена программного продукта рассчитывается по формуле 9:

$$\text{Ц} = \text{З}_{\text{итого}} + \Phi_{\text{отр}} * 0,3; \quad (9)$$

Тогда,  $\text{Ц} = 17008,3 + 1210 * 0,3 = 17371,3$  руб

Разработка веб-ресурса преследовала следующие цели:

- 1) Обеспечить взаимодействия авторов и издательства;
- 2) Рассчитать стоимость незапланированных печатных изданий;
- 3) Обеспечить слепое рецензирование;
- 4) Облегчить подачу материала в издательство.

Дополнительные доходы от реализации веб ресурса выражаются в увеличении количества незапланированных печатных изданий. Так если каждый месяц будет печататься 2 незапланированных издания, с коррективками, в размере 5 штук, объемом 5 страниц, то это составит 60391,18.

$\text{Д доп} = 60391,18 - 13990,32 = 46400,86$  рублей

Экономическая эффективность находится по формуле 10:

$$\text{Э (П)} = \text{Д доп} - \text{Ц} = 46400,86 - 17371,3 = 29029,56 \text{ рублей}; \quad (10)$$

Таким образом экономический эффект рассчитывается по формуле 11:

$$\text{Э(ф)} = \text{Э(П)} / \text{Ц} = 29029,56 / 17371,3 = 1,67; \quad (11)$$

Рентабельность проекта находится по формуле 12:

$$\text{Р} = \text{Э(П)} / \text{Ц} * 100 = 29029,56 / 17371,3 * 100 = 167 \%; \quad (12)$$

Срок окупаемости рассчитывается по формуле 13:

$$\text{Ток} = \text{Ц} / \text{Э(П)} = 17371,3 / 29029,56 = 0,60 \text{ или } 219 \text{ дней}; \quad (13)$$

Из данных расчетов можно сделать вывод, что разработка веб ресурса была экономически выгодна и оправдана. Основные технико-экономические данные представлены в таблице 6.

Таблица 6 – Основные технико-экономические показатели проекта

| Наименование работ               | Основные характеристики               | Единицы измерения | Проект   |
|----------------------------------|---------------------------------------|-------------------|----------|
| Разработка технического задания  | Трудоемкость выполняемых работ        | чел./час.         | 495      |
| Планирование web-сайта           | Себестоимость проекта                 | руб.              | 17008,3  |
| Рабочее проектирование web-сайта | Цена программного продукта(web-сайта) | руб.              | 17371,3  |
|                                  | Дополнительные доходы                 | руб.              | 46400,86 |
| Отладка и тестирование web-сайта | Экономическая эффективность (прибыль) | руб.              | 29029,56 |
|                                  | Экономический эффект                  | раз.              | 1,67     |
| Обобщение и оценка результатов   | Рентабельность                        | (%)               | 167      |
|                                  | Срок окупаемости проекта              | дни.              | 219      |

Таким образом, в результате расчетов были получены следующие данные:

- 1) На разработку приложения было затрачено 17371,3 рублей.
- 2) Дополнительные доходы составили 46400,86 рублей.
- 3) Экономическая эффективность 29029,56 рублей, что в 1,67 раз превышает затраты на создание продукта.
- 4) Рентабельность проекта составили 167 %.

Вывод: создание рабочего веб приложения довольно длительный процесс. Необходимо провести более тщательное тестирование и доработку всех выявленных недостатков. Так же данный проект имеет большое функционала, который еще можно внедрить для улучшения работы веб ресурса.

Внедрение данной системы в организацию является целесообразным решением. Уже через 219 дней вложения полностью себя окупят.

## ЗАКЛЮЧЕНИЕ

В наше время вопрос, связанный с автоматизацией производства путем внедрения информационных систем наиболее актуальный. Информационные системы повышают производительность труда, упрощают ведение и корректировку учета материалов и выпускаемой продукции.

Выпускная квалификационная работа представляет собой описание проектирования и разработки веб ресурса. Веб ресурс решает следующие задачи:

1. Обеспечивает взаимодействие авторов и издательства;
2. Позволяет рассчитать стоимость незапланированных печатных изданий;
3. Обеспечивает слепое рецензирование;
4. Облегчает подачу материала в издательство.

В ходе выполнения работы была изучена деятельность ПГУ им. Т.Г. Шевченко, его организационная структура, история и расчеты стоимости печатного издания.

В среде разработки Visual Studio и Visual Studio Code был разработан и реализован веб ресурс. Для работы было использовано множество дополнительных библиотек, а также работа с SQL.

Была создана документация в виде Swagger для дальнейшего использования и обновления API.

Расчёт экономической эффективности веб ресурса показал, что затраты, окупаются меньше чем за год.

Таким образом, следует считать, что задачи выпускной квалификационной работы полностью выполнены и цель исследования достигнута.

2Казакова,1 самарин



## СПИСОК ЛИТЕРАТУРЫ

1. Boris Cherny Programming TypeScript: Making Your JavaScript Applications Scale / O'Reilly Media; 1-е издание 2019 – 324с.
2. David Flanagan JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language / O'Reilly Media; 7th edition 2020 – 706с.
3. JavaScript на примерах/ Никольский А. П. - СПб.: Наука и Техника, 2017. - 272 с.
4. Баженова, И.Ю. Языки программирования: Учебник / И.Ю. Баженова. - М.: Академия, 2018
5. Бурашников Д.А., Хлыст Д.А. Разработка web-сайта на основе html с использованием javascript / ФГБОУ ВО «Кубанский государственный технологический университет» 2019 – 65-74с.
6. В.В. Дабежа. Е.А. Матвейчук, Издание кафедральных работ ПГУ им. Т.Г. Шевченко / Научно методическое пособие 2011г.
7. В.И. Казакова, Е.А. Зайцева Издательская деятельность вузов: моменты инерции / Нижегородский государственный технический университет ИМ. Р.Е. АЛЕКСЕЕВА 2013- 45-52 с.
8. Гайкова П.Д. Возможности субд ms sql server и языка программирования с# по шифрованию данных в информационной системе "Автостоянка" конференция - информационная безопасность: современная теория и практика 2019 – 28-33с.
9. Грофф Джеймс Р., Вайнберг Пол Н. SQL. Полное руководство / Вильямс -Полный справочник, 2018 - 980с.
- 10.И.Г. Семакин, А.П. Шестаков. Основы алгоритмизации и программирования. – М.: Академия, 2008. – 400 с.
- 11.Иванов А.Н., Шегай Л.Ю. Метод разработки веб-приложения с использованием современных технологий на основе языка программирования javascript / Тихоокеанский государственный университет, Хабаровск 2020 – 267-274с.

- 12.Кириченко А.В., Хрусталеv А.А. HTML+CSS3.Основы современного web-дизайна – СПб «Наука и Техника» 2018г.
- 13.Кобылкин Д.С., Юсупова О.В. Перспектива применения средств entity framework при разработке систем распределенной обработки данных на языке C# / ФГБОУ ВО «Оренбургский государственный университет» 2021 – 14-17с.
- 14.Лашина М.В., Соловьев Т.Г. Информационные системы и технологии в экономике и маркетинге. Учебное пособие. – М.: КноРус, 2019. – 304 с.
- 15.Листуон Бенджамин, Хэнчетт Эрик Vue.js в действии / Питер, среия – библиотека программиста 2019 – 3004с.
- 16.М.В. Коломейчук, А.Н. Федоренко, О.А. Штырова Электронные издания. Правила оформления ГОСТ 7.0.83-2013 Тирасполь 2019.
- 17.П.Б. Хорев. Объектно-ориентированное программирование. – М.: Академия, 2012. – 448 с.
- 18.План издания печатных работ пгу им. Т.г. Шевченко в 2019 г.
- 19.Положение об издательской деятельности Приднестровского Государственного Университета им. Т.г. Шевченко, Система менеджмента и качества /Тирасполь 2017
- 20.Положение об Издательстве Приднестровского университета им. Т.Г. Шевченко Тирасполь 2017г.
- 21.Поляков О.В. Практическое руководство по составлению бизнес-плана и оценке эффективности инвестиционного проекта / О.В. Поляков. - М.: ИЛ, 2019.
- 22.Ричард Стоунз, Нейл Мэттью. PostgreSQL. Основы. – М.: Символ-Плюс, 2002. – 640 с.
- 23.Родионов В. В., Воронкин Е. Ю. Разработка web-приложения для оптимизации организационных задач на языке программирования c# asp.netcore / Сибирский государственный университет геосистем и технологий, Россия, г. Новосибирск , 2021 – 160-163с.

- 24.С.А. Войтенкова, О.А. Штырова, Л.В. Савицкая, А.А. Михайленко, А.Н. Федоренко Правила набора текстового материала учебных изданий / Инструктивно-методические указания, Тирасполь 2014г.
- 25.Сизова О. В., Пляцев И. Ю. Разработка требований к информационной системе учёта операционной деятельности типографии / ФГБОУ ВО «Ивановский государственный химико–технологический университет», 2019 – 101-104 с.
- 26.Стилмен Эндрю, Грин Дженнифер Head First. Изучаем C# / Издательство: Питер, Серия: Head First (O'Reilly) 2022 - 768с.
- 27.Ю. Н. Самарин. Полиграфическое производство: учебник для вузов /— 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2019. — 497 с.
28. .NET 6 [Электронный ресурс]: <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6>
- 29.SQL в примерах [Электронный ресурс]: <https://www.w3schools.com/sql/>
- 30.Введение в REST API — RESTful веб-сервисы [Электронный ресурс]: <https://habr.com/ru/post/483202/>
- 31.Официальный сайт Typescript [Электронный ресурс]: <https://www.typescriptlang.org/>
- 32.Официальный сайт Vue3 [Электронный ресурс]: <https://vuejs.org/>
- 33.Приложение для создания макетов. [Электронный ресурс]: <https://Draw.io>
- 34.Регулярные выражения, пособие для новичков [Электронный ресурс]: <https://habrahabr.ru/post/115825/>
- 35.Электронная библиотека Elibrary [Электронный ресурс]: <https://elibrary.ru/>

# ПРИЛОЖЕНИЕ

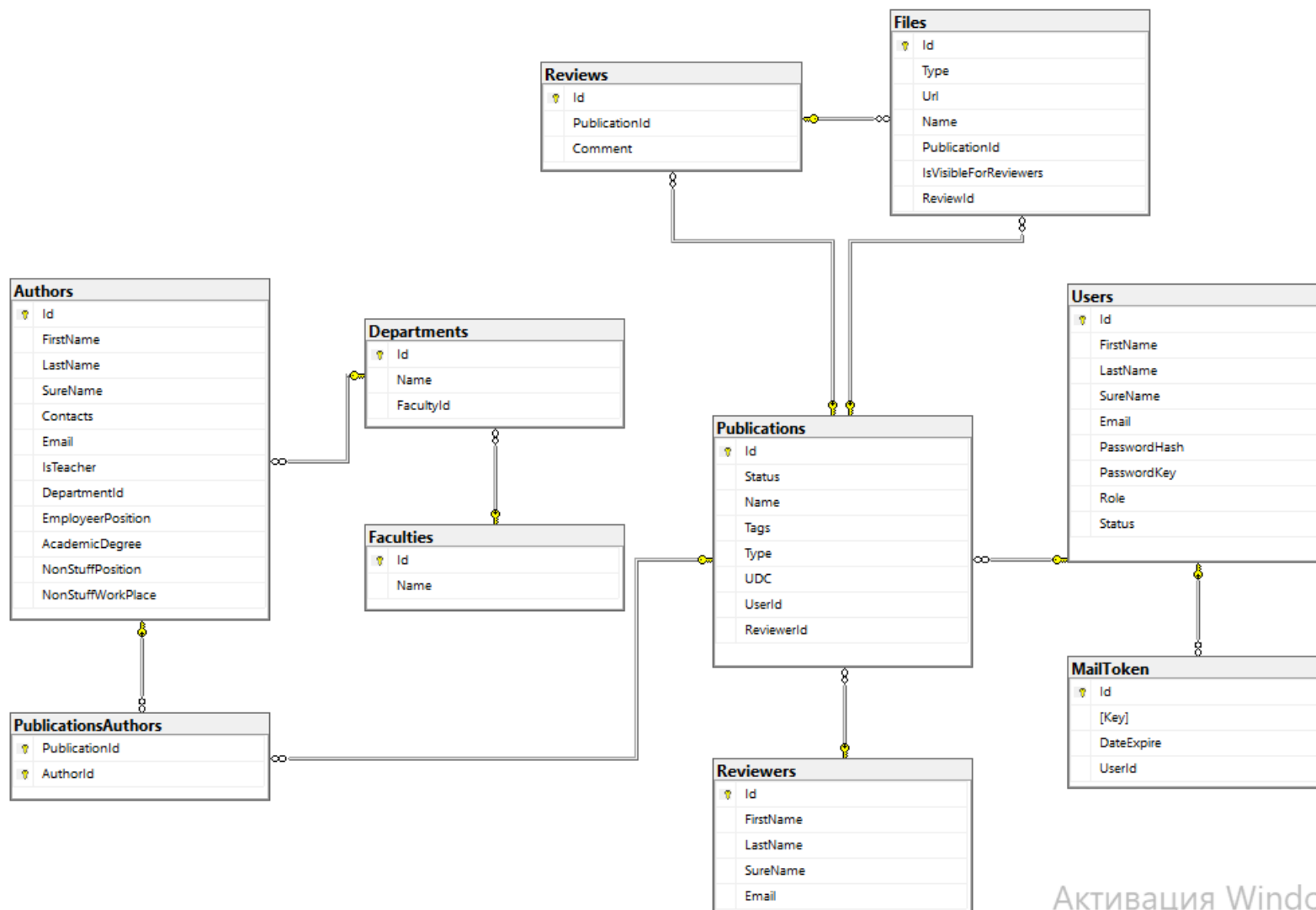
## Приложение А

```
public class PublicationAuthorsConfiguration :
IEntityTypeConfiguration<PublicationAuthors>
{
    public void
Configure(EntityTypeBuilder<PublicationAuthors> builder)
    {
        builder.HasKey(x => new {x.AuthorId,
x.PublicationId});

        builder
            .HasOne(x => x.Author)
            .WithMany(x => x.Publications)
            .HasForeignKey(x => x.AuthorId)
            .IsRequired(true);

        builder
            .HasOne(x => x.Publication)
            .WithMany(x => x.Authors)
            .HasForeignKey(x => x.PublicationId)
            .IsRequired(true);
    }
}
```

## Приложение Б



## Приложение В

```
[HttpPost]
[Authorize]
[Route($"{nameof(Add)}")]
[ProducesResponseType(200, Type =
typeof(BaseResponse<long>))]
[ProducesResponseType(400, Type =
typeof(BaseResponse))]
public async Task<BaseResponse<long>>
Add([FromBody] AddPublicationModel model)
{
    var result = await
_publication.AddPublicationAsync(model);
    return new BaseResponse<long>(result);
}
```

## Приложение Г

```
public async Task<long>
AddPublicationAsync (AddPublicationModel model)
{
    var publication = new Publication
    {
        Name = model.Name,
        Status = EnumPublicationStatus.Draft,
        Tags = model.Tags,
        Type = model.Type,
        UDC = model.UDC,
        UserId = model.UserId
    };

    await _db.Publications.AddAsync (publication);
    await _db.SaveChangesAsync();
    return publication.Id;
}
```

## Приложение Д

```
class HttpResponseResult<T>{  
    isSuccess: boolean  
    errorMessage: string  
    stackTrace: string  
    TypeName?: string  
    data: T  
}
```



## Приложение Е

```
class PublicationService extends BaseRequestService{
    constructor(api: WebApiService) {
        super(api, "Publication");
    }

    public async Add(request:
AddPublicationRequestModel):
Promise<HttpResponseResult<any>> {
        let res = await
this.createRequestPromise<any>('/Add', 'POST',
request);
        return res;
    }

    public async Get(request:
GetPublicationRequestModel):
Promise<HttpResponseResult<GeneralModel<Array<GetPublic
ationResponseModel>>>> {
        let res = await
this.createRequestPromise<any>('/Get', 'GET', request);
        return res;
    }

    public async SetStatus(request:
SetStatusPublicationRequestModel):
Promise<HttpResponseResult<any>> {
        let res = await
this.createRequestPromise<any>('/SetStatus', 'PATCH',
request);
        return res;
    }

    public async Update(request:
UpdatePublicationRequestModel):
Promise<HttpResponseResult<any>> {
        let res = await
this.createRequestPromise<any>('/Update', 'PATCH',
request);
        return res;
    }
}
```

## Приложение Ж

```
class BaseRequestService{

    protected _controllerName:string;
    protected _api:WebApiService;
    constructor(api:WebApiService,
controllerName:string) {
        this._api = api;
        this._controllerName = controllerName;
    }

    protected    createRequestPromise<T>(url:string,
method:string,                requestBody?:          any):
Promise<HttpResponseResult<T>>
    {
        let self = this;
        return                                          new
Promise<HttpResponseResult<T>>((resolve) => {

self._api.sendXHR<T>(`/${self._controllerName}${url}`,m
ethod,requestBody, (res)=>{
            resolve(res)
        });
    });
    }
}
```

## Приложение И

```
public sendXHR<T>(url: string, method: string, data?: any,
callback?: (res: HttpResponseResult<T>) => void): void {
    // построение записи
    var self = this;
    if (!url)
        url = ''
    if (!url.startsWith('http')) {
        if (!url.startsWith('/'))
            url = '/' + url;
        url = this.baseUrl + url;
        if (data && !url.includes("Add") &&
!url.includes("UploadFileForPublication") ||
url.includes("PublicationAuthor/Add")) {
            let params: string = ""
            let array: Array<string> = []
            for (let i = 0; i < Object.keys(data).length; i++) {
                let key = Object.keys(data)[i]
                if (typeof data[key] == "object") {
                    let testkey = key
                    let el = data[key]
                    Object.keys(el).map(key => {
                        if (el[key] != "")
                            array.push(this.buildString(key, el[key],
testkey))
                    })
                } else if (data[key] != "")
                    array.push(this.buildString(key, data[key]))
            }
            params = array.join("&")
            url = url + "?" + params
        }
    }
    let xhr = new XMLHttpRequest();
    xhr.onload = function () {
        let resp: HttpResponseResult<T>;
        try {
            resp = JSON.parse(xhr.response)
        }
        catch {
            resp = xhr.response
        }
        let res: HttpResponseResult<T> = {
            isSuccess: resp.isSuccess,
            TypeName: resp.TypeName,
            data: resp.data || null,
            errorMessage: resp.errorMessage,
            stackTrace: resp.stackTrace
        }
        if (xhr.status != 200 && xhr.status != 201) { // анализируем
HTTP-статус ответа, если статус не 200, то произошла ошибка
            console.error(self.TAG, 'Request:', url, xhr.status,
xhr.statusText);
            if (xhr.status === 401) {
                Cookies.set(AuthService.AdminAuthTokenName, '')
                setTimeout(() => {
                    location.reload();
                }, 1000);
            }
        }
    }
}
```

```

        }, 1000);
    }
    if (callback)
        callback(res);
} else { // если всё прошло гладко, выводим результат
    let headers = {}
    xhr.getAllResponseHeaders().split('\r\n').forEach(x => {
        let header = x.split(':').map(x => x.trim());
        headers[header[0]] = header[1];
    })
    if (callback)
        callback(res);
}
};

xhr.onerror = function (e) { // происходит, только когда запрос
    совсем не получилось выполнить
    // alert(`Ошибка соединения`);
    let res: HttpResponseResult<T> = {
        isSuccess: false,
        errorMessage: "Ошибка при отправке запроса",
        data: null,
        stackTrace: ""
    }
    if (callback)
        callback(res);
};

xhr.onprogress = function (event) { // запускается периодически
    // event.loaded - количество загруженных байт
    // event.lengthComputable = равно true, если сервер присылает
заголовок Content-Length
    // event.total - количество байт всего (только если
lengthComputable равно true)
    //alert(`Загружено ${event.loaded} из ${event.total}`);
};

xhr.open(method, url, true);
let jwtoken = Cookies.get(AuthService.AdminAuthTokenName);
if (jwtoken)
    xhr.setRequestHeader('Authorization', jwtoken);
let body:any
if (typeof data === 'string') {
    body = data;
}
else if (data) {
    body = new FormData();
    this.buildFormData(body, data);
    body = JSON.stringify(data)
}
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.send(body);
}

```

## Приложение К

```
import { createRouter, createWebHistory, RouteRecordRaw, RouterView } from "vue-router";
const BaseTemplate = () => import("@views/layouts/base-layout.vue");
const AdminLayout = () => import("@views/layouts/admin/admin-layout.vue");

import AuthorsPage from "@views/pages/admin/authors/authors.vue"
import AuthorsDetailed from "@views/pages/admin/authors/authors-detailed.vue"
import MaterialsAdminPage from "@views/pages/admin/materials/materials.vue"
import Reviewer from "@views/pages/admin/reviewer/reviewer.vue"
import ReviewerDetailed from "@views/pages/admin/reviewer/reviewer-detailed.vue"
import MaterialDetailed from "@views/pages/admin/materials/material-detailed.vue"
import NewMaterials from "@views/pages/admin/new-materials/new-materials.vue"
import NewMaterialDetailed from "@views/pages/admin/new-materials/new-material-detailed.vue"
import Departments from "@views/pages/admin/departments/departments.vue"
import { ADMIN, AUTHORS, DETAILEDADMIN, ADMINMATERIALS, ALLADMINMATERIALS, NEWMATERIALADMIN, ALLNEWMATERIALADMIN, DETAILEDNEWMATERIALADMIN, REVIEWER, ALLREVIEWER, REVIEWERDETAILED, AUTHOR, AUTHORSGROUP, DEPARTMENTS } from "../routerNames";

//#endregion
const AdminRouter: Array<RouteRecordRaw> = [
  {
    path: '/',
    component: BaseTemplate,
    children: [
      {
        name: ADMIN,
        path: ADMIN,
        component: AdminLayout,
        redirect: { name: NEWMATERIALADMIN },
        children: [
          {
            name: AUTHORSGROUP,
            path: AUTHORSGROUP,
            component: BaseTemplate,
            redirect: { name: AUTHORS },
            children: [
              {
                name: AUTHORS,

```

```

        path: AUTHORS,
        component: AuthorsPage,
        meta: { title: "Авторы" }
    },
    {
        name: AUTHOR,
        path: AUTHOR,
        component: AuthorsDetailed,
        meta: { title: "Детальная автора" }
    },
]
},
{
    name: REVIEWER,
    path: REVIEWER,
    component: BaseTemplate,
    redirect: { name: ALLREVIEWER },
    children: [
        {
            name: ALLREVIEWER,
            path: ALLREVIEWER,
            component: Reviewer,
            meta: { title: "Рецензент" }
        },
        {
            name: REVIEWERDETAILED,
            path: REVIEWERDETAILED,
            component: ReviewerDetailed,
            meta: { title: "Детальная рецензента" }
        },
    ]
},
{
    name: ADMINMATERIALS,
    path: ADMINMATERIALS,
    component: BaseTemplate,
    redirect: { name: ALLADMINMATERIALS },
    children: [
        {
            name: ALLADMINMATERIALS,
            path: ALLADMINMATERIALS,
            component: MaterialsAdminPage,
            meta: { title: "Материалы" }
        },
        {
            name: DETAILEDADMIN,
            path: DETAILEDADMIN,
            component: MaterialDetailed,
            meta: { title: "Детальная материала" }
        },
    ]
},
},

```

```

    {
      name: NEWMATERIALADMIN,
      path: NEWMATERIALADMIN,
      component: BaseTemplate,
      redirect: { name: ALLNEWMATERIALADMIN },
      children: [
        {
          name: ALLNEWMATERIALADMIN,
          path: ALLNEWMATERIALADMIN,
          component: NewMaterials,
          meta: { title: "Новые материалы" }
        },
        {
          name: DETAILEDNEWMATERIALADMIN,
          path: DETAILEDNEWMATERIALADMIN,
          component: NewMaterialDetailed,
          meta: { title: "Детальная нового материала" }
        }
      ],
    },
  ],
},
{
  name: DEPARTMENTS,
  path: DEPARTMENTS,
  component: Departments,
  meta: { title: "Факультеты и кафедры" }
},
],
},
];

export default AdminRouter;

```

## Приложение Л

```
const install = (app: App, opts: { router: Router }) =>
{
  opts.router.beforeEach(async (to, from, next) => {

    const cookie =
Cookies.get(AuthService.AdminAuthTokenName);
    const role = Cookies.get(AuthService.Role);

    let isAuthorized = !!cookie && cookie !== ''
    if (to.name === INDEX || to.name === CALCULATION ||
to.name === LOGIN) {
      next()
      return
    }
    if (isAuthorized && from.name !== LOGIN) {
      if (role === UserRole.Admin &&
to.fullPath.includes("user")) {
        next({ name: ADMIN });
        return;
      }
      else if (role === UserRole.User &&
to.fullPath.includes("admin")) {
        next({ name: USER });
        return;
      }
    }
    if (to.name === LOGIN && isAuthorized) {
      if (role === UserRole.Admin) {
        next({ name: ADMIN });
        return;
      }
      else if (role === UserRole.User) {
        next({ name: USER });
        return;
      }
    }
    if (!to.meta.allowAnonymous && !isAuthorized) {
      next({ name: INDEX })
      return
    }
    next()
  });
}
export default { install,};
```



## Приложения М

```
<template>
  <div class="content-block">
    <div class="content-block-all">
      <div class="content-block-title">
        
        <div v-if="title" class="content-title">{{ title }}</div>
      </div>
      <slot name='header' />
    </div>
    <slot />
  </div>
</template>
<script lang="ts">
import { Options, Vue, Prop } from "vue-property-decorator";

@Options({
  name: "content",
})
export default class ContentComponent extends Vue {
  @Prop({ type: String, default: "" }) title: string;
  @Prop({ type: Boolean, default: false }) isBack: boolean;
}
</script>
<style lang="less" scoped>
.content-block {
  padding: 14px 26px;
  .content-block-all {
    display: flex;
    justify-content: space-between;
    .content-block-title {
      display: flex;
      text-align: center;
      padding-bottom: 12px;
      .arrow-back {
        margin-right: 10px;
      }
      .content-title {
        font-size: 17px;
        font-weight: 600;
        border-bottom: 2px #f0f0f0 solid;
      }
    }
  }
}
}
</style>12
```

## Приложение Н

### Сравнительный анализ по характеристикам web-ресурса

| Характеристика   | Пояснение | Tipar -<br>типография -<br><a href="https://tipar.ru/">https://tipar.ru/</a> | Printstore Group -<br><a href="https://printstore.com.ua/">https://printstore.com.ua/</a> / | Перо -<br><a href="https://peroprint.ru/">https://peroprint.ru/</a> | Artpoligraf -<br><a href="https://artpoligraf.md/">https://artpoligraf.md/</a> |
|--|-----------|--|---|---|--|
| Дизайн страниц   |           |  |   |   |  |
| Выдержаны ли цвета, шрифты, графика в едином стиле?  |           | 2  | 1   | 2   | 2  |
| Сбалансированы ли цвета дизайна страниц?   |           | 2  | 2   | 2   | 2  |
| Не перегружена ли страница информацией (особенно касается главных страниц)?                                      |           | 2  | 1   | 2   | 2  |
| Качественна ли графика и сочетается ли она с остальными составляющими страницы?                                  |           | 2  | 1   | 2   | 2  |
| Не мешает ли графика пользователю воспринимать информацию?   |           | 2  | 1   | 2   | 2  |
| Удобство использования   |           |  |   |   |  |
| Насколько проста и понятна навигация?  |           | 2  | 2   | 2   | 2  |
| Позволяет ли навигация вернуться на предыдущие подуровни?  |           | 2  | 0   | 2   | 2  |
| Приходится ли пользователю делать более 3х кликов для того, чтобы найти нужную информацию?                       |           | 2  | 1   | 2   | 2  |
| Присутствуют ли на страницах сайта объяснения или подсказки для пользователя (пункты меню навигации, поля форм)? |           | 1  | 0   | 2   | 2  |
| Работает ли сайт на всех платформах и браузерах?   |           | 2  | 2   | 2   | 2  |
| Структура и навигация  |           |  |   |   |  |

|  |    |    |    |    |
|--|----|----|----|----|
| Понятно ли как попасть на каждую страницу?                               | 2  | 2  | 2  | 2  |
| Логотип ведет на главную страницу?                                       | 2  | 2  | 2  | 2  |
| Присутствует поиск по страницам?   | 2  | 0  | 0  | 2  |
| Меню закрепляется при прокрутке страницы?                                | 2  | 0  | 0  | 0  |
| Можно легко и быстро найти нужную страницу?                              | 2  | 1  | 2  | 2  |
| Полнота контента   |    |    |    |    |
| Ясно ли предназначение сайта?  | 2  | 2  | 2  | 2  |
| Присутствует ли информация на всех страницах (во всех разделах) сайта?   | 2  | 1  | 2  | 2  |
| Ориентирован ли сайт на местную интернет-аудиторию?                      | 2  | 2  | 2  | 2  |
| Соответствует ли содержание сайта (текстовое, графическое) его тематике? | 2  | 2  | 2  | 2  |
| Отсутствуют грамматические или синтаксические ошибки?                    | 2  | 2  | 2  | 2  |
| Наличие обратной связи   |    |    |    |    |
| Присутствует ли обратная связь на сайте?                                 | 2  | 2  | 2  | 2  |
| Присутствует ли блок «контакты» с соответствующей информацией            | 2  | 2  | 2  | 2  |
| Присутствуют ли адреса   | 2  | 2  | 2  | 2  |
| Присутствует ли мобильный телефон  | 2  | 2  | 2  | 2  |
| Присутствуют ли ссылки на социальные сети                                | 2  | 2  | 2  | 2  |
| Итого  | 49 | 35 | 46 | 48 |

## Приложение П

### Сравнительный анализ по организационно-экономическим показателям

|  |  |   |  |   |
|--|--|---|--|---|
| Характеристика                                     | Tipar типография - <a href="https://tipar.ru/">https://tipar.ru/</a>                       | Printstore Group - <a href="https://printstore.com.ua/">https://printstore.com.ua/</a> /  | Перо - <a href="https://pero-print.ru/">https://pero-print.ru/</a>   | Artpoligraf - <a href="https://artpoligraf.md/">https://artpoligraf.md/</a>   |
| География аудитории                                | Нет данных   | Нет данных  | Нет данных   | Нет данных  |
| Индексация сайта<br>Яндекс<br>Google               | Яндекс Проиндексировано 62 <sup>-61</sup><br>Яндекс ИКС 10<br>Google: Проиндексировано 124 | Яндекс Проиндексировано 747<br>Яндекс ИКС 40<br>Google: Проиндексировано 688  | Яндекс Проиндексировано 80<br>Яндекс ИКС 370<br>Google: Проиндексировано 899 <sup>+44</sup>  | Яндекс Проиндексировано 72 <sup>-45</sup><br>Яндекс ИКС 10<br>Google: Проиндексировано 504 <sup>-22</sup>                 |
| Время загрузки страниц                             | 1.36 с   | 2.8 с.  | 0.11 с.  | 1.53 с.   |
| Ссылки   | Уникальных ссылок 1<br>Домены 1<br>IP-адреса 1<br>Подсети 1                                | Уникальных ссылок 36<br>Домены 20<br>IP-адреса 19<br>Подсети 19   | Уникальных ссылок 493<br>Домены 308<br>IP-адреса 80<br>Подсети 75  | Уникальных ссылок 65<br>Домены 15<br>IP-адреса 15<br>Подсети 13   |
| Время пребывания на сайте                          | Нет данных   | Нет данных  | 00:00:52   | Нет данных  |
| Ключевые слова                                     | типар пмр, типография тирасполь, типография пмр, типар бендеры, зао типар                  | print store, printstore, printstore group, доставка печатной продукции, принт стопе, online print store, store print, печать стпрес | издательство перо москва, издательский дом перо, перо издательство сайт, издательство перо официальный сайт, ооо издательство перо | Artpoligraf, carti de vizita la comanda Chisinau, creare cartele de fidelitate, printer pret moldova, xerox scan chisinau |
| Видимость сайта                                    | 0 показов за месяц   | 0 показов за месяц  | 629 показов за месяц   | 0 показов за месяц  |
| Популярность сайта (хосты – уникальные посетители) | Нет данных   | Глобальный рейтинг 7,171,505  | Глобальный рейтинг 2,546,793   | Глобальный рейтинг 3,372,688  |