МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)
Энергетический факультет
Кафедра информатики, вычислительной техники и прикладной математики

ОТЧЕТ

по производственной практике (технологической (проектно-технологической))
(вид практики)

на (в) ООО «ТВОЙ РОБОТ»
(место прохождения практики, указать полное наименование организации)

обучающегося Борисовой Екатерины Олеговны
(имя, отчество, фамилия)

Курс 3 Группа ИВТ-20

Направление подготовки: 09.03.01 Информатика и вычислительная техника
(код, наименование)

Направленность ОП Программное обеспечение вычислительной техники и автоматизированных систем

Руководитель практики от университета:

доцент _____ /Валова Ольга Валерьевна
(должность) (подпись) (имя, отчество, фамилия)

Руководитель практики от предприятия:

Руководитель отдела разработки _____ /Яценко Андрей Владимирович
(должность) (подпись) (имя, отчество, фамилия)
(печать организации)

Чита

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)
Энергетический факультет
Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на производственную практику (технологическую (проектно-технологическую))

Студенту: Борисовой Екатерины Олеговны

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Индивидуальное задание на практику:

1. Создание базы данных для приложения.
2. Создание панели редактора.
3. Обоснование выбора средств реализации.
4. Программная реализация.
5. Написание и защита отчета по практике.

Дата выдачи задания 03.07.2023 г.

Руководитель от университета_____/Валова О. В.

(подпись, расшифровка подписи)

Задание принял к исполнению 03.07.2023 г.

Подпись студента_____/Борисова Е. О.

(ФИО)

Содержание

ВВЕДЕНИЕ

Краеведческий туризм – это уникальный вид туризма, который позволяет путешественникам погрузиться в историю и культуру мест, которые посещают. В отличие от обычного туризма, краеведческий туризм подразумевает изучение и понимание местных традиций, обычаев и достопримечательностей, посещение природных достопримечательностей.

Одной из главных причин, почему краеведческий туризм является важным, является его способность сохранять и передавать историю и культуру регионов. Путешественники, знакомясь с историческими местами и местными традициями, помогают сохранить наследие прошлого и продолжают его традиции. Кроме того, такой вид туризма способствует развитию региональной экономики, создавая рабочие места и увеличивая поток туристов.

Краеведческий туризм также способствует здоровому образу жизни. Путешествия в места с историческим и культурным наследием часто включают пешие прогулки и активные виды отдыха, такие как велосипедные прогулки, походы, сплавы по рекам или альпинизм. Такие деятельности помогают укрепить физическую форму и повысить уровень активности, что положительно сказывается на здоровье.

В нашем крае нет единого источника, в котором были бы указаны возможные для посещения места, о многих из них жители нашего края даже и не знают. Таким образом онлайн-гид, на котором можно будет узнать об этих местах, посмотреть маршрут, найти попутчиков, быть может выбрать турагентство, которое предложит многодневный поход или конную прогулку, будет очень актуален и востребован.

1 Обоснование выбора средств реализации

Для разработки онлайн-гида необходимы:

1) фронтенд веб-сайт;
2) база данных;
3) бэкенд веб-сайт;
4) сервер, на котором будут храниться сайт.

Для разработки фронтенда веб-сайта используются язык разметки HTML, язык стилей CSS и язык программирования JavaScript [1]. Эти языки позволяют задавать внешний вид страницы и обеспечивать правильное отображение в популярных браузерах. Для работы с кодом можно использовать простой текстовый редактор, такой как Блокнот, или более удобные редакторы, например, VS Code или Sublime Text, которые облегчают работу с кодом, подсвечивают синтаксис и указывают на ошибки.

Перед началом верстки создается макет сайта, который помогает визуализировать будущий дизайн. Для этой цели хорошо подходит Figma, где можно создать черновой или более подробный макет, продумать расположение элементов интерфейса и выбрать цветовую гамму.

Данные о пользователях, материалах сайта, бронировании экскурсий и рассылках хранятся в базе данных, которую можно администрировать с помощью СУБД SQLite [2]. Эта система обладает не только простотой использования, но и встроенными механизмами безопасности, возможностью выполнения операций CRUD и использования SQL для запросов данных из JSON-документов, составления отчетов и статистики. Для визуального отображения таблиц и связей между ними можно использовать Draw.io, в котором доступен весь нужный функционал [3].

Для разработки бэкенда веб-сайта будет использоваться фреймворк Laravel. Laravel – бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller – модель-представление-контроллер). Laravel

выпущен под лицензией MIT [4]. Для работы с этим фреймворком так же можно использовать VS Code.

Чтобы сайт был доступен в браузере, его необходимо подключить к серверу. Хостинг – это услуга, при которой клиент арендует часть или весь сервер у хостинг-провайдера. Любой сайт – это набор текстовых и видеофайлов, изображений, кода, баз данных и прочей информации. Чтобы люди могли увидеть их, сайт должен быть загружен на компьютер– со специальным программным обеспечением и круглосуточным доступом в интернет. Такой компьютер называется сервером. У компании-хостера достаточно серверов для обслуживания огромного количества клиентов [5]. Если хочется, можно организовать хостинг на домашнем компьютере, но для этого нужен достаточно мощный компьютер и стабильное Интернет-соединение [6]. Существуют различные сервисы, предоставляющие услуги хостинга на разных условиях. Также можно обойти проблему с сервером, используя хранилище на GitHub [7]. В этом случае файлы сайта хранятся в удаленном репозитории, что позволяет не беспокоиться о потере данных и о заполнении места на сервере.

2 Программная реализация

2.1 Создание макета сайта

Важным этапом создания сайта является проработка идеи и создание макета, который пригодится в дальнейшей вёрстке. В этом нам поможет редактор Figma, имеющий широкий перечень возможностей.

Черновой макет – схематичный набросок, выполненный для понимания того, как на сайте будут располагаться основные информационные блоки, графика и прочие элементы дизайна.

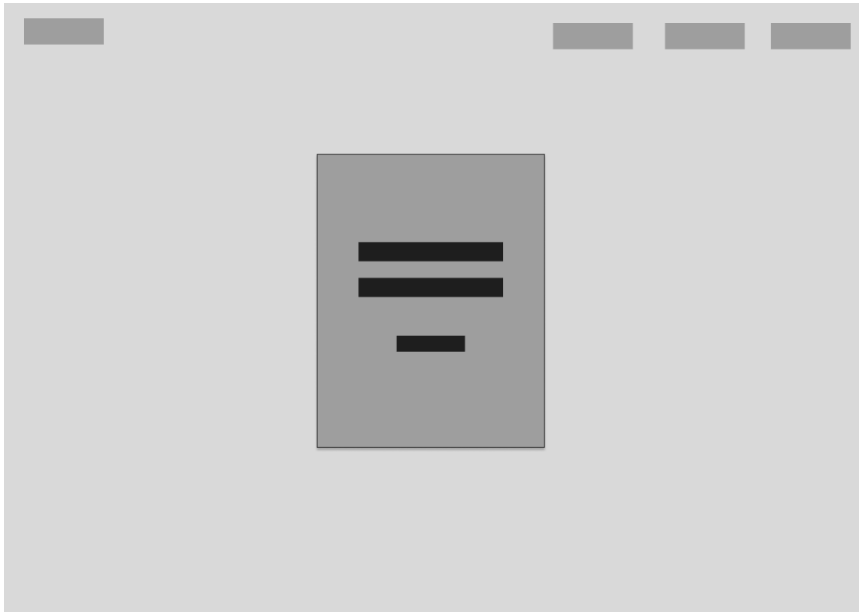

Рисунок 2.1 – Черновой макет сайта

После чернового макета можно приступать к более детальному. Для подбора цветовой палитры используем ресурс Encycolopedia [8]. Здесь можно посмотреть различные цветовые вариации, схемы и многое другое.

Рисунок 2.2 – Интерфейс Encycolopedia

Опираясь на подобранную цветовую палитру и черновой макет, строится более детальная схема. Здесь используются уже не только геометрические фигуры, дизайн усложняется.

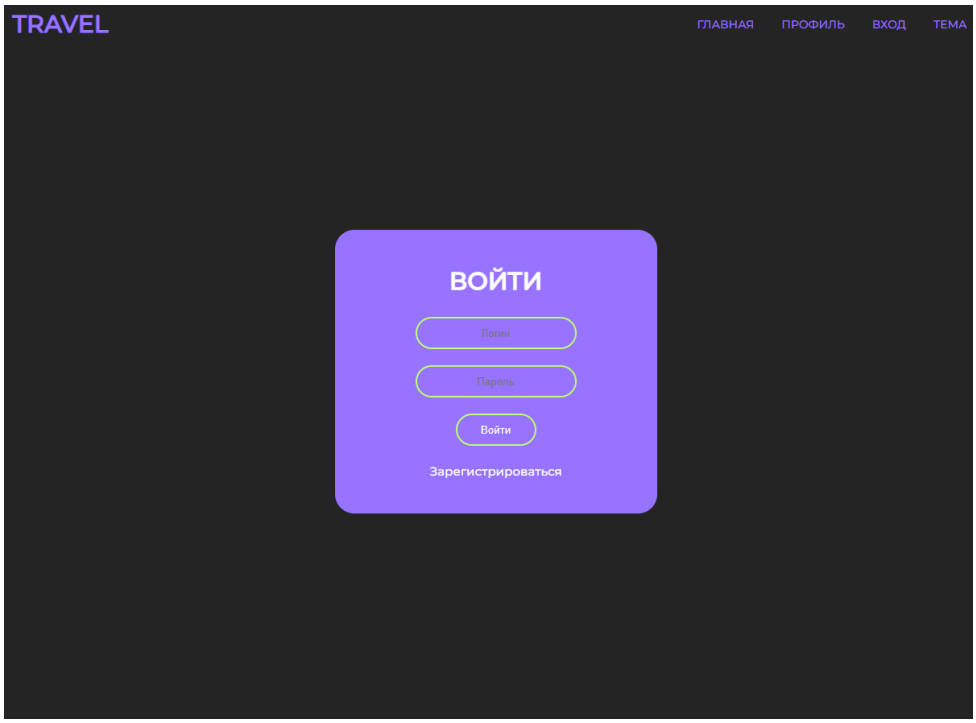

Рисунок 2.3 – Детальный макет сайта

Затем пишется код HTML. При помощи стилей CSS можно придать дизайну объёма и сложности, а также упростить работу, когда нужно использовать много одинаковых по оформлению элементов.

2.2 База данных

База данных в онлайн гиде нужна для следующих целей:

1) хранение данных о пользователях сайта;
2) хранение данных о природных достопримечательностях;
3) хранение данных о туристических организациях;
4) запись (бронь) на походы;
5) учет пользователей, подписавшихся на новостную рассылку.

В данной диаграмме наглядно видно каждый элемент базы данных и его связь с другими элементами. Используя данную схему можно визуализировать работу базы данных.

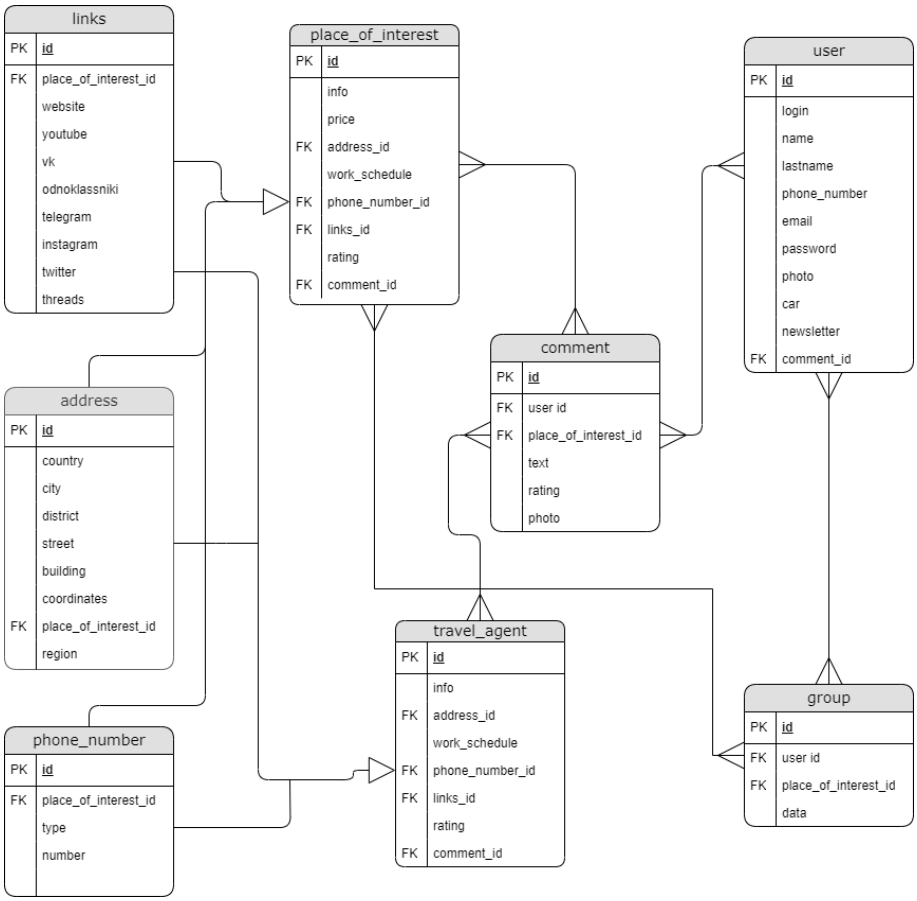

Рисунок 2.4 – Связи таблиц БД (физическая модель)

2.3 Бэкенд веб-сайта

После создания ER-модели можно приступить к созданию базы данных. В Laravel база данных создается через миграции [9].

Далее нужно создать панель редактора для возможности добавления публикаций в базу данных, она будет доступна по адресу: http://127.0.0.1:8000/admin.

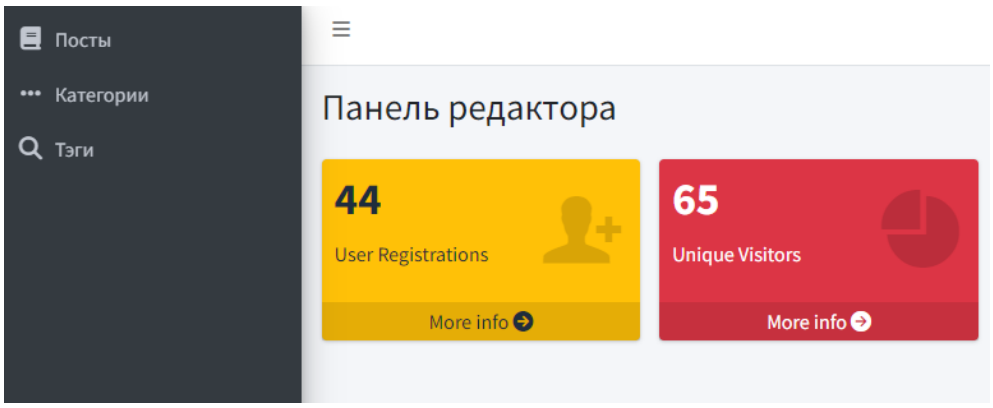

Рисунок 2.5 – Панель редактора

Также необходимо реализовать страницы просмотра, редактирования и добавления категорий, тэгов и публикаций.

Для реализации функционала этих страниц необходимо написать контроллеры для категорий, тэгов и публикаций соответственно [10]. Эти контроллеры будут позволять добавлять, редактировать и удалять данные из базы данных, а также выводить их на страницу.

Контроллер категорий:

```
class CategoryController extends Controller
{
    public function index()
    {
        $categories = Category::all();

        return view('admin.category.index',
compact('categories'));
    }

    public function create()
    {
        return view('admin.category.create');
    }

    public function delete(Category $category)
    {
        $category->delete();
        return redirect()->route('admin.category.index');
    }

    public function edit(Category $category)
    {
        return view('admin.category.edit',
compact('category'));
    }

    public function show(Category $category)
```

```
        {
            return view('admin.category.show',
compact('category'));
        }

        public function store(StoreRequest $request)
        {
            $data = $request->validated();
            Category::firstOrCreate(['title' => $data['title']]);
            return redirect()->route('admin.category.index');
        }

        public function update(UpdateRequest $request, Category
$category)
        {
            $data = $request->validated();
            $category->update($data);
            return view('admin.category.show',
compact('category'));
        }
    }
```

Контроллер тэгов:

```
    class TagController extends Controller
    {
        public function create()
        {
            return view('admin.tag.create');
        }

        public function delete(Tag $tag)
        {
            $tag->delete();
            return redirect()->route('admin.tag.index');
        }

        public function edit(Tag $tag)
        {
            return view('admin.tag.edit', compact('tag'));
        }

        public function index()
        {
            $tags = Tag::all();

            return view('admin.tag.index', compact('tags'));
        }

        public function show(Tag $tag)
        {
            return view('admin.tag.show', compact('tag'));
        }
```

```
        public function store(StoreRequest $request)
        {
            $data = $request->validated();
            Tag::firstOrCreate(['title' => $data['title']]);
            return redirect()->route('admin.tag.index');
        }

        public function update(UpdateRequest $request, Tag $tag)
        {
            $data = $request->validated();
            $tag->update($data);
            return view('admin.tag.show', compact('tag'));
        }
    }
```

Контроллер для публикаций:

```
    class PostController extends Controller
    {
        public function create()
        {
            $categories = Category::all();
            $tags = Tag::all();
            return view('admin.post.create',
compact('categories', 'tags'));
        }

        public function delete(Post $post)
        {
            $post->delete();
            return redirect()->route('admin.post.index');
        }

        public function edit(Post $post)
        {
            $categories = Category::all();
            $tags = Tag::all();
            return view('admin.post.edit', compact('post',
'categories', 'tags'));
        }

        public function index()
        {
            $posts = Post::all();

            return view('admin.post.index', compact('posts'));
        }

        public function show(Post $post)
        {
            return view('admin.post.show', compact('post'));
        }

        public function store(StoreRequest $request)
```

```
        {
            try{
                $data = $request->validated();

                $tagIds = $data['tag_ids'];
                unset($data['tag_ids']);
                //заносим в бд путь к изображению (Storage::put
сохраняет изображение и возвращает путь к нему)
                $data['preview_image'] = Storage::disk('public')-
>put('/images', $data['preview_image']);
                $data['main_image'] = Storage::disk('public')-
>put('/images', $data['main_image']);

                $post = Post::firstOrCreate([
                    'title' => $data['title'],
                    'content' => $data['content'],
                    'preview_image' => $data['preview_image'],
                    'main_image' => $data['main_image'],
                    'category_id' => $data['category_id'],
                ]);
            $post->tags()->attach($tagIds);
            } catch(\Exception $exseption){
                abort(404);
            }

            return redirect()->route('admin.post.index');
        }

        public function update(UpdateRequest $request, Post
$post)
        {
            $data = $request->validated();
            $tagIds = $data['tag_ids'];
            unset($data['tag_ids']);
            //заносим в бд путь к изображению (Storage::put
сохраняет изображение и возвращает путь к нему)
            $data['preview_image'] = Storage::disk('public')-
>put('/images', $data['preview_image']);
            $data['main_image'] = Storage::disk('public')-
>put('/images', $data['main_image']);


            $post->update($data);
            $post->tags()->sync($tagIds);

            return view('admin.post.show', compact('post'));
        }
    }
```

2.4 Запуск сайта

Для запуска сайта нужно не только позаботиться о дизайне и наполнении, но подумать еще и о хостинге. Как упоминалось ранее, существует несколько вариантов:

1) арендовать хостинг у провайдера;

2) использовать в качестве сервера домашний ПК;

3) использовать в качестве сервера репозиторий на GitHub.

Последний способ является наиболее выигрышным, так как не требуются никакие финансовые вложения, к тому же, это достаточно просто осуществить. Достаточно создать репозиторий с названием name.github.io, где name – название сайта, загрузить туда файл index.html, содержащий код главной страницы, и сайт уже готов к работе и его можно посмотреть не только с личного компьютера, но и с других устройств.

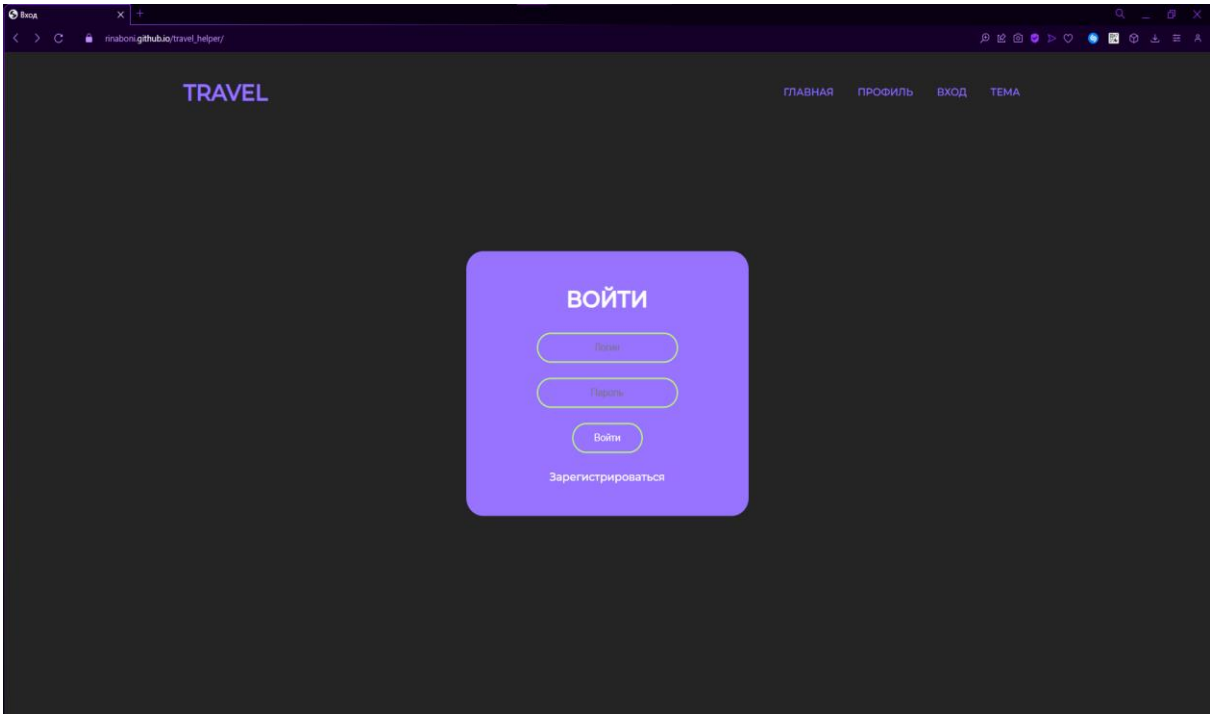

Рисунок 1.5 – Запущенный сайт

ЗАКЛЮЧЕНИЕ

Результатом проделанной работы является создание страниц входа и регистрации, панель редактора, с помощью которой возможно создание публикаций, так же была реализованная связь с базой данной.

В дальнейшем планируется реализация возможности комментирования публикация, создание публикаций пользователем, возможность для пользователей собираться в группы для путешествия.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Что такое верстка сайта: виды, методы, инструменты. – Текст: электронный // Школа онлайн обучения программированию Loftschool. – 2022. – URL: https://loftschool.com/blog/posts/verstka-sajta-chto-eto (дата обращения: 01.07.2023).

2. SQLite – замечательная встраиваемая БД (часть 1). – Текст: электронный // Хабр. – 2012. – URL: https://habr.com/ru/articles/149356/ (дата обращения: 01.07.2023).

3. Entity Relationship Diagram (ERD). – Текст: электронный // draw.io: официальный сайт. – 2016. – URL: https://drawio-app.com/blog/entity-relationship-diagram-erd/ (дата обращения: 01.07.2023).

4. Laravel. – Текст: электронный // Википедия свободная энциклопедия. – 2023. – URL: https://ru.wikipedia.org/wiki/Laravel (дата обращения: 01.07.2023).

5. Создание сайта без платных хостингов. – Текст: электронный // Хостинговое сообщество «Timeweb Community» – база знаний о хостинге и серверных технологиях. – 2020. URL: https://timeweb.com/ru/community/articles/kak-sdelat-hosting-doma (дата обращения: 01.07.2023).

6. Как сделать сервер дома. – Текст: электронный // Skillbox Media — журнал для профессионалов. Актуальные статьи про бизнес, дизайн, образование, разработку игр и программирование. – 2020. – URL: https://skillbox.ru/media/marketing/chto-takoe-khosting-dlya-sayta-i-kak-ego-vybrat/ (дата обращения: 01.07.2023).

7. GitHub Pages. – Текст: электронный // GitHub Pages: официальный сайт. – 2023. – URL: https://pages.github.com/ (дата обращения: 01.07.2023).

8. Encycolopedia. – Текст: электронный // Hex Color Codes, Paint Matching and Color Picker.– 2023. – URL: https://encycolorpedia.com (дата обращения: 01.07.2023).

9. Database: Migrations. – Текст: электронный // Laravel - The PHP Framework For Web Artisans. – 2023. – URL: https://laravel.com/docs/10.x/migrations (дата обращения: 01.07.2023).

10. Controllers. – Текст: электронный // Laravel - The PHP Framework For Web Artisans. – 2023. – URL: https://laravel.com/docs/10.x/controllers (дата обращения: 01.07.2023).