Министерство образования и науки Челябинской области  
государственное бюджетное профессиональное образовательное учреждение  
«Южно-Уральский государственный колледж»

ПЦК «Информационных технологий»

ДОПУЩЕН К ЗАЩИТЕ

Председатель ПЦК «Информационных технологий»

Н.А.Назарова

« » 20\_\_ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка информационного сайта новостей

специальность 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Выпускную работу выполнил: студент группы ИС448Д,  очного отделения | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. К. Черняев |
| Руководитель: преподаватель ПЦК «ИТ» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. М. Костенко |
| Рецензент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

Челябинск 2021

Министерство образования и науки Челябинской области  
государственное бюджетное профессиональное образовательное учреждение  
«Южно-Уральский государственный колледж»

ПЦК «Информационных технологий»

УТВЕРЖДАЮ

Председатель ПЦК «Информационных технологий»

Н.А.Назарова

« » 20\_\_ г.

ЗАДАНИЕ  
на выполнение выпускной квалификационной работы

Студент группы ИС448Д очной формы обучения

Черняев Константин Константинович

Тема работы Разработка информационного сайта новостей

утверждена приказом по колледжу № 743/у от 15.12.2020 года

Сроки сдачи студентом законченной работы «14» июня 2021 года

1. Исходные данные к работе
2. Разработать информационный сайт новостей

Перечень вопросов, подлежащих разработке:

* изучение принципов работы и построения веб-сайтов;
* сбор материалов;
* анализ полученной информации;
* выбор технологии и программных средств для реализации продукта;
* руководство программиста;
* наполнение сайта информацией;
* руководство пользователя;
* экономическая часть;
* техника безопасности при работе с ПК.

Перечень приложений, схем, плакатов

Приложение А, Б, В

Календарный график:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этапы работы | Характер работы | Руководитель Ф.И.О. | Срок выполнения | Подпись, дата | |
| Задание выдал | Задание принял |
| I | Выбор и уточнение темы | И. М. Костенко | Декабрь, 2020 |  |  |
| II | Составление развернутого плана | И. М. Костенко | Январь, 2021 |  |  |
| III | Сбор, изучение, систематизация теоретических источников и фактического материала | И. М. Костенко | Февраль-март, 2021 |  |  |
| IV | Предоставление первого варианта | И. М. Костенко | Апрель, 2021 |  |  |
| V | Предоставление окончательного варианта | И. М. Костенко | Май, 2021 |  |  |
| VI | Направление на внешнее рецензирование | И. М. Костенко | Июнь, 2021 |  |  |

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_/ И. М. Костенко

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_/ К. К. Черняев

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г

ПОЯСНИТЕЛЬНАЯ ЗАПИСКАСОДЕРЖАНИЕ

Д

Изм.

*Лист*

*№ докум.*

*Подпись*

*Дата*

Лист

4

ПЗ – ЮУГК – 090207 – 2021

Разраб.

Черняев К. К.

Провер.

Костенко И. М.

Реценз.

.

Н. Контр.

Костенко И. М.

Руководитель

Утверд.

Назарова Н.А.

Разработка   
информационного сайта новостей

Лит.

Листов

59

*ПЦК ИТ ИС448Д*

[**СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ** 7](#_Toc75010269)

[**ВВЕДЕНИЕ** 8](#_Toc75010270)

[1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 10](#_Toc75010271)

[1.1 Анализ поставленной задачи 10](#_Toc75010272)

[1.2 Выбор метода создания сайта 10](#_Toc75010273)

[2. **ОСНОВНАЯ ПРОЕКТНАЯ ЧАСТЬ** 16](#_Toc75010277)

[2.1 Сценарий разработки сайта 16](#_Toc75010278)

[2.2 Алгоритм разработки сайта 17](#_Toc75010279)

[2.3 Руководство программиста 23](#_Toc75010290)

[2.4 Руководство пользователя 29](#_Toc75010295)

[**ЗАКЛЮЧЕНИЕ** 33](#_Toc75010303)

[**БИБЛИОГРАФИЧЕСКИЙ СПИСОК** 34](#_Toc75010304)

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

ОРМ — преобразователь объектных отношений базы данных в объект структуры.

Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

VPS или VDS, виртуальный выделенный сервер — услуга предоставления в аренду так называемого виртуального выделенного сервера.

SQL – язык запросов в базу данных.

ВВЕДЕНИЕ

В рамках выпускной квалификационной работы было изучено множество сайтов по схожей тематике. В ходе исследования была выявлена явная перегруженность интерфейса, мелкий и сложно читаемый шрифт, множество визуальных и информационных излишеств, а также множество контента, который может вызвать отрицательные эмоции или волнение у посетителей этих сайтов.

Конечным продуктом ВКР является «лёгкая» альтернатива вышеупомянутым сайтам, контент на котором будет содержать лишь художественные и позитивные новости, в связи с чем были ограничены категории постов.

Компании из различных отраслей проводят исследования на тему того, как конкретно скорость загрузки влияет на отток аудитории. Так, по данным компании Akamai, ведущего поставщика услуг для акселерации веб-сайтов, 40% посетителей уходят с сайта, если он загружается более трех секунд. А один из крупнейших ритейлеров мира Walmart выяснил, что у сайта магазина есть всего 0,25 секунды, чтобы удержать посетителя или он перейдет на сайт конкурента. В связи с этим была использована архитектура сайта SPA (Single Page Application), так как при ее использовании нет необходимости перезагружать страницу, а также используются легковесные JavaScript и CSS библиотеки.

Целью ВКР является создание информационной системы, которая позволит просматривать, и создавать информационные посты с новостями.

Задачи ВКР:

* Анализ предметной темы;
* Создание плана разработки;
* Изучение инструментов разработки;
* Разработка дизайн-концепта сайта;
* Разработка базы данных и информационных моделей;
* Программирование;
* Наполнение контентом;
* Тестирование;

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ
   1. Анализ поставленной задачи

Сайт, или веб-сайт — одна или несколько логически связанных между собой веб-страниц[5]. Новостной сайт — это интернет-издание, специализация которого заключается в сборе и выдаче общетематических новостей или новостных материалов на одну тему[6].

Новостные сайты могут принадлежать СМИ, которые изначально не специализировались на интернет-публикациях (АиФ, Forbes, Комсомольская правда и т. д.) или же быть независимыми проектами.

На новостном сайте обязательно должна быть лента новостных постов. Посты должны разделяться на категории, соответственно должно быть меню фильтрации. На сайте обязательно должен быть поиск по новостным постам.

В проекте требуется реализовать дополнительные функциональные возможности для зарегистрированных пользователей, такие как добавление в избранное, персонализированная лента новостей и создание постов.

* 1. Выбор метода создания сайта

Создание веб-приложения – бывает очень трудоемким, и чтобы упросить разработку и проектирование были разработаны различные программные решения и библиотеки. Каждое из них подходит для определенной задачи в большей или меньшей степени. Требуется рассмотреть варианты различных методов разработки с использованием дополнительного ПО и выбрать из них наиболее подходящий.

Графические редакторы

Для выполнения поставленной задачи требуется создать дизайн-макет, чтобы представить, как будет выглядеть сайт. Для выполнения этой задачи были рассмотрены следующие графические редакторы:

CorelDraw – бесплатный векторный редактор, с помощью которого можно создать сайт, используя примитив. В редакторе присутствуют инструменты для выравнивания, группировки, смешивания цветов, присутствует множество эффектов, имеется возможность редактирования векторных фигур по точкам.

Adobe Illustrator – проприетарный векторный графический редактор, имеющий множество инструментов для создания эскизов, кадрирования, цветокоррекции. Функционально мало чем отличается от CorelDraw, но удобство использования заметно выше.

Figma – удобный визуальный редактор, созданный для прототипирования сайтов. В ней сочетается простота и функциональность. Главные особенности этого приложения:

* можно работать как в веб странице, так и в приложении;
* синхронизация проектов через облако;
* компоненты;
* автоматическое выравнивание (auto layout);
* условная бесплатность;
* генерация css стилей.

Из всех претендентов наиболее подходящим является Figma из-за ее направленности на создание дизайна сайтов, удобства интерфейса и кроссплатформенности.

IDE

Для удобства написания кода рекомендуется использовать интеллектуальную среду разработки (IDE).Есть три главных конкурента в этой сфере.

Первый это Sublime **–** условно бесплатный легкий текстовый редактор. Сам по себе он не является полноценным IDE, но имеет достаточно большой функционал по сравнению с обычным блокнотом. Этот редактор имеет более 40 поддерживаемых языков программирования с неплохим автодополнением. Опционально можно установить дополнительные пакеты расширений для конкретного языка или программы в общем.

Второй претендент – Visual Studio Code (VS Code). В отличии от Sublime почти не имеет встроенных функций для автодополнения и валидации кода. Эти функции добавляются через магазин дополнений.

Последний - Intellij PHPStorm - IDE для создания веб приложений, в частности на языках PHP и JavaScript. Имеет множество функций включая работу с базой данных, пакетными менеджерами и сборщиками. Это приложение индексирует все файлы в проекте и на основе полученных данных предлагает варианты автодополнения.

PHPStorm имеет в себе все нужные функции для создания веб приложений, а если чего-то не хватает, имеется магазин плагинов. Эта IDE создана для разработки на PHP. На основе этого она была выбрана как основная среда разработки.

Фреймворк

Разработчики выбирают фреймворки, потому что они ускоряют разработку, обладают безопасностью «из коробки» и способны масштабироваться довольно легко. ДляPHP обычно используют один из трех фреймворков: Laravel, Yii2 и Symfony. Ниже перечислены их плюсы и минусы.

Laravel.

Плюсы:

* удобная система миграций;
* подробная и понятная документация;
* интегрированная система модульного тестирования;
* встроенный шаблонизатор blade;
* очень гибкое формирование перенаправлений;
* очень гибкие возможности для написания rest api;
* очень быстро развивается;
* много документации на любую тему;
* популярен во всем мире;

Минусы:

* большой функционал работает через фасады, и ide-системы не видят методов и свойств в некоторых классах, показывая предупреждения;
* изучается немного сложнее yii2;

Yii2.

Плюсы:

* визуальный генератор кода;
* интуитивная mvc-архитектура;
* не требователен к ресурсам;
* подходит для проектов любой сложности и масштаба.

Минусы:

* не очень гибкое формирование перенаправлений;
* слишком склеенные библиотеки для frontend'а с backend'ом.

Symfony.

Плюсы:

* подробная и понятная документация;
* достаточно высокая скорость работы ядра;
* низкая связность компонентов.

Минусы:

* сложный и очень ресурсоемкий ОРМ;
* высокий порог входа;
* перенасыщенность разного рода сущностями;
* интегрированный аннотационный синтаксис.

По причине популярности, удобства и не высокого порога вхождения был выбран фреймворк Laravel. Так же у Laravel есть экосистема микрофреймворков и библиотек, их называют «стек TALL», что расшифровывается как «TailwindCSS AlpineJS Laravel Livewire».

Tailwind CSS – легковесная CSS библиотека, которая реализовывает концепцию «Utility-First», а именно создание сложных компонентов из множества примитивных утилит (классов). В этой библиотеке используются различные классы, которые часто изменяют небольшое количество CSS атрибутов. К примеру класс «flex» применяет к контейнеру стиль «display:flex», a mx-auto выставляет отступы по Х на значение auto. Это может показаться странным, писать все стили прямо в HTML, но потом понимаешь, что это очень удобно, а главное производительно, особенно учитывая, что у этой библиотеки есть модификаторы адаптивности и состояния, например ‘md:’ или ‘hover:’ . Эта библиотека написана на JavaScript, поэтому легко настраивается и адаптируется под задачи. К примеру, можно изменить ширину селекторов или «зеленый» цвет, как в моем случае. Используя postCSS можно создавать свои классы основываясь на существующих, что позволяет спрятать слишком объемные стили в CSS, применив их к новому классу с помощью директивы @apply после чего будет сгенерирован статический .css файл. Это далеко не весь функционал, который использовался в этой работе.

Alpine JS – легкая библиотека, подключающаяся через CDN, из-за чего совершенно не влияет на загрузку страницы. Эта библиотека позволяет взаимодействовать с dom-элементами, как с компонентами, использовать переменные, управлять состоянием, создавать анимации и многое другое с помощью HTML атрибутов.

Livewire – является чем-то между приложением на NodeJS и динамическими страницами PHP. С помощью JavaScript этот модуль взаимодействует с бэкендом и делает это без особых усилий со стороны программиста. Достаточно изящное решение проблемы спагетти из JavaScript’ов, так как всё управление происходит посредством HTML атрибутов и Event’ов.

Так же в проекте используются две библиотеки «CKEditor 5» и «Axios».

CKEditor 5 [25], [26], [27] – достаточно мощный модульный текстовый редактор для текста в браузере. Им пользуются тысячи компаний, и он давно зарекомендовал себя для этой цели. Из плюсов – простота установки и обширные возможности. Этот плагин является бесплатным для коммерческого и некоммерческого использования.

Axios – инструмент для выполнения HTTP запросов на сервер. В отличии от fetch, axios имеет более обширный и удобный функционал. В нем имеются удобные функции разных видов запроса (POST, GET, DELETE, PUT, UPDATE) и сразу же писать обработчик для данных.

В наше время почти каждый проект имеет базу данных для хранения данных приложения и пользователей. Чтобы управлять потоками данных требуются производительные СУБД. Так как данные будут структурированы и часто требовать сортировки и выборки потребуется реляционная база данных. Для вебсайта обычно выбирают либо MySql (SqlLite), либо Postgresql. По причине более высокой производительности был выбран Postgres.

Из-за высокой стоимости хостинга и невозможности установки дополнительного ПО (например, Postgesql) будет использован VPS.

После выбора средств и методов разработки можно приступить к практике и начать с составления сценария разработки сайта.

1. ОСНОВНАЯ ПРОЕКТНАЯ ЧАСТЬ
   1. Сценарий разработки сайта

Создавать сайт лучше всего, когда есть дизайн-макет. По этой причине сначала нужно создать его. Помимо этого, для разработки базы данных можно построить диаграмму связей, по которой будет удобно ориентироваться в структуре таблиц.

Разрабатываемый программный продукт будет основываться на фреймворке Laravel. Он будет отвечать за маршрутизацию, обработку информации, управление базой данных, загрузку и передачу файлов, построение страницы для отправки клиенту, API для frontend скриптов и другое. Для обновления данных на странице будет использоваться Livewire, с его помощью будет построена логика взаимодействия с клиентом, например, переход по страницам, редактирование и просмотр данных из базы данных. Для добавления динамичности страницам будет использоваться AlpineJS, он будет отвечать за взаимодействия с клиентом, которые не требуют загрузки данных с сервера, например, анимаций меню или всплывающие окна.

Взаимодействие с базой данных будет происходить через систему миграций и «Eloquent» ORM Laravel. Миграции позволят легко развернуть приложение на сервере и обновлять структуру базы данных без использования импорта SQL файлов. Так же миграции универсальны, и будет не сложно сменить СУБД в будущем и перенести структуру в нее.

После создания сценария разработки сайта можно приступать к самой разработке.

* 1. Алгоритм разработки сайта

Дизайн

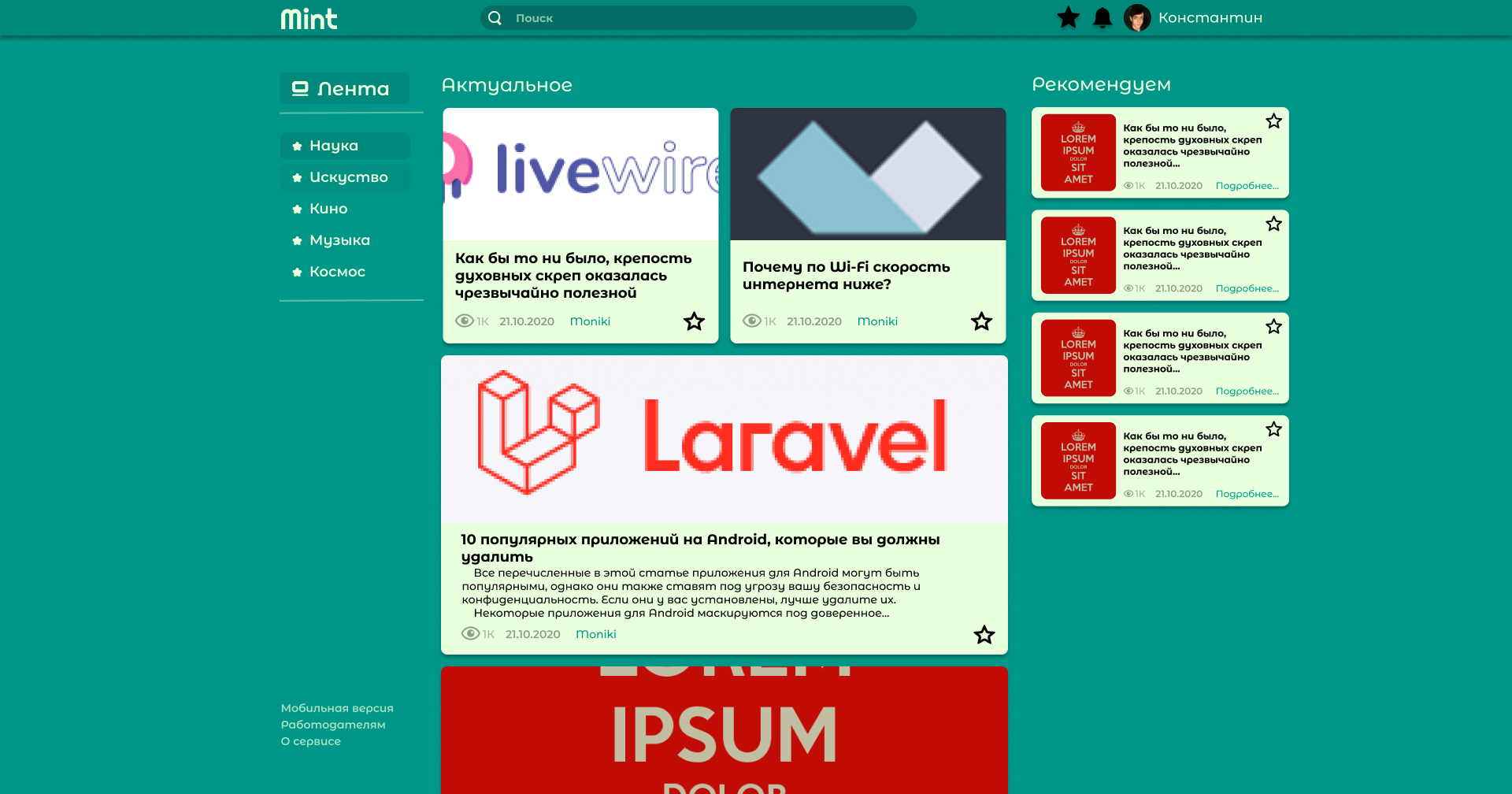
****

Рисунок 1 - Дизайн главной страницы

Для начала следует создать дизайн сайта (результат работы изображен на рисунке 1). В качестве цветового оформления фона и элементов сайта был выбран голубовато-зеленый цвет и два его более темных оттенка. Текст имеет две окраски – черный и светло-зеленый. В дизайне не использовался белый цвет, вместо него выступает бледно зеленый, близкий к цвету свечения белого фосфора. Такое сочетание не так сильно контрастирует, чем не напрягает глаза читателя.

Шрифт на всех страницах сайта «Montserrat Alternates». Этот шрифт не имеет засечек, достаточно гладкий, что достаточно приятно выглядит в окружении скругленных форм элементов сайта. Этот шрифт нельзя изменить в посте, ведь одна из главных особенностей дизайна сайта – это его однородность. При переходе между страницами пользователь не должен замечать явных изменений в дизайне, поэтому шрифт и его вариации зафиксированы.

Анимации

Для придания живости сайта было сделано несколько анимаций, вот самые явные:

* Всплывающее окно регистрации
* Проявляющееся меню в шапке сайта
* Плавно выпадающий список результатов запроса
* Анимация загрузки страницы
* Анимация появления меню (мобильная версия сайта)
* Отклик

Очень важно, чтобы пользователь понимал, с какими элементами он может взаимодействовать. Для этого используется стиль графического дизайна «Material design» [28], разработанный компанией Google. Добавив к элементам тени и изменяя их цвет, форму и заполнение при наведении. Чтобы сконцентрировать внимание пользователя на всплывающих окнах, фон становится более темным, а при нажатии вне активного окна – оно тут же закроется и фон исчезнет. Выбранная категория подсвечивается темным оттенком, а при наведении на другую категорию она лишь немого потемнеет.

Структура сайта

Для упрощения ознакомления пользователя с интерфейсом сайта был использован стандартный макет социальной сети. Когда пользователь заходит на сайт ему уже знакомо большинство элементов интерфейса. Слева расположены категории, под которыми располагаются дополнительные ссылки, справа актуальные новости, сверху шапка с логотипом, ведущим на главную страницу, поиском и различные кнопки меню. По центру расположен основной контент страницы – лента, пост, страница настроек или создания поста. Структура сайта позволяет менять лишь центральную часть, и не загружать статические элементы несколько раз.

Адаптивность

Для поддержки разных диагоналей экрана было принято решение скрывать элементы в порядке их полезности от меньшего к большему. К примеру, при уменьшении окна браузера менее 1500px исчезает правая часть с рекомендациями, а ее место занимает центральный блок. Когда экран сужается до 1200px исчезают отступы с краев, что позволяет комфортно читать заголовки новостей. При уменьшении ширины до мобильной исчезает меню категорий и появляется кнопка для его отображения, меню пользователя сворачивается до одной иконки, любые надписи пропадают, а центральная часть начинает занимать всё свободное место. Все посты, кроме тех, что в правом ряду адаптируются под новую ширину, изменяя набор элементов, их размеры и положение.

Виды отображения постов и их различия.

Основной любого поста является его превью, название, дата публикации и количество просмотров. Каждый пост можно добавить в избранное, для этого предусмотрен переключатель в виде звезды. Всего на сайте 4 вида отображения поста:

* Пост в поиске – небольшой по высоте, адаптивный
* Пост в правой колонке – фиксированная ширина и высота
* Пост в ленте – большой, занимает всю ширину, по высоте регулируется заголовком, ограничен по высоте
* Полностраничный пост (для просмотра) – занимает всю центральную зону.

Пост состоит из множества компонентов, которые переиспользуются в каждом из них. Так можно сэкономить время на переписывании определенных элементов по всему коду.

Структура базы данных

За основу была взята классическая архитектура блога, после чего добавлялись дополнительные таблицы и связи (схема структуры базы данных находится в приложении А). Для создания отношений были использованы отношения моделей Laravel[15], для этого требуется соблюсти некоторые правила наименования полей и таблиц. Чтобы упростить разработку и поддержку, база данных была построена с помощью миграций[24]. В базе данных присутствует несколько таблиц, связывающих пользователя и пост, они служат разным целям и нужны лишь для сохранения отношения этих двух моделей. Присутствует структура для работы с учетными записями, восстановлением пароля, подтверждением почты и другими функциями.

Верстка

Для верстки сайта использовался синтаксис blade компонентов [19].

Шапка сайта содержит в себе 3 основных блока – логотип, поиск и кнопки меню. Шапка содержит отступы с двух сторон, максимальная ширина шапки 1200px, что позволяет взаимодействовать с ней даже на очень широких мониторах без долгих перемещений мыши.

Справа располагаются кнопки меню. Если пользователь не авторизован, то кнопка одна – вход, иначе их 2. Кнопка с рисунком звезды отрывает ленту с постами, которые находятся в закладках, а нажатие на аватар пользователя открывает меню, в котором есть возможность выйти, создать пост, перейти в настройки профиля и перейти на вкладку помощь.

Поиск располагается посередине, там, где его ожидает увидеть пользователь. Если в поле поиска вставить поисковой запрос, то экран затемнится, произойдет запрос на сервер[21], тот вернет результаты, после чего они плавно отобразятся, «вылетая» сверху страницы. Если запрос не дал результата, то будет выведено сообщение об этом. При снятии фокуса с поля поиска и результатов затемнение пропадает, а найденные посты скрываются. Когда фокус возвращается, посты снова отображаются без запроса на сервер

Блок актуальных новостей

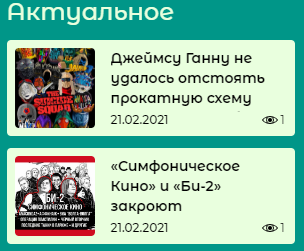
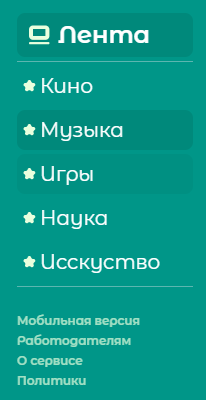
В этот блок актуальных новостей выводятся последние новости, добавленные на сайт. Он исчезает при уменьшении окна браузера до определенной ширины. Скрытие контролируется AlpineJS с помощью тега x-show. Внешний вид блока изображен на рисунке 2.

Рисунок 2 - Блок актуальных новостей

Левое **боковое** меню

Боковое меню является компонентом AplineJS. Для вывода списка категорий используется шаблон, содержащий параметры запроса, который будет происходить при нажатии и имя категории. При нажатии на категорию происходит изменение состояния приложения для применения фильтра. Кнопка «Лента» снимает фильтрацию и изменяет текущую страницу на ленту новостей. Под списком категорий находятся низкоприоритетные ссылки. Это меню сворачивается при достижении минимальной ширины экрана, кроме того, в шапке появляется иконка открытия меню. При нажатии на эту кнопку блок меню становится видимым, и его ширина вытесняет основной блок, а не наплывает на него, что позволяет не затенять фон меню, а значит не нужно видоизменять цветовую палитру кнопок. Внешний вид меню категорий изображен на рисунке 3.

Рисунок 3 - меню   
категорий

Лента новостей

Лента новостей обновляется по мере добавления постов. Для зарегистрированных пользователей выводится индивидуальная лента новостей, основанная на лайках пользователя. При нажатии на звезду в шапке происходит фильтрация по избранному пользователя. Под лентой новостей находятся кнопки пагинации. Все элементы кроме ленты зафиксированы, из-за чего прокручивается только она. Полоса прокрутки специально стилизована.

Страница настроек

На странице настроек пользователя можно изменить параметры для учетной записи пользователя, пароль и посмотреть статистику. Присутствует возможность загружать аватар пользователя. Изменения на странице обрабатываются Livewire. Поля автоматически обновляются на сервере при отправке любого запроса, например, запроса на изменение пароля. Все данные проверяются на сервере, после чего приходит ответ об успехе или отмене операции. В обоих случаях пользователь получает информационное сообщение. Исключением являются чек боксы, они обновляются без уведомлений.

* 1. Руководство программиста

Алгоритм перехода по страницам

Так как сайт задумывался как SPA, частые перезагрузки не могут быть допущены, но в то же время нельзя просто изменять части страницы без возможности перейти к той или иной части приложения по прямой ссылке. Чтобы решить эту проблему используется технология истории из History API. С помощью скрипта при нажатии кнопки добавляется новая запись в историю браузера, после чего отправляется Event, который перехватывается Livewire, а тот в свою очередь изменяет содержимое страницы. Но в этой схеме не решается вопрос с прямой ссылкой. Чтобы его решить в контроллере главной странице было прописано условие, которое проверяет, с какого адреса поступил запрос, и если у пользователя есть доступ к этой странице, изменяет ее, в ином случае – загружает ленту. Чтобы добавить возможность перемещаться по страницам с помощью кнопок вперед и назад был создан JavaScript слушатель, который отправляет запрос на изменение страницы при передвижении по истории.

Страница создания поста

На странице создания поста можно в реальном времени посмотреть, как будет выглядеть пост на сайте, так как все поля ввода не имеют фона, а структура документа полностью повторяет структуру опубликованного поста.

Для изменения заголовка вводной части используется тег <textarea> с прозрачным фоном. Эти поля не имеют рамок и не позволяют пользователю использовать переносы строк и вставлять их. При переносе строки блок ввода расширяется по вертикали на одну строку. Поля никнейм пользователя и дата публикации заполняются автоматически. Чтобы не стилизовать кнопку ввода изображения для всех браузеров, было принято решение заменить её на картинку, показывающую размер изображения после загрузки. При нажатии на эту картинку открывается окно выбора изображения с помощью вызова события «click» у спрятанного элемента «input». После загрузки изображения в браузер с помощью JavaScript браузер получает ссылку на это изображение, и эта ссылка помещается в тег <img>. Так отображается превью изображение поста в реальном времени. При повторном нажатии происходят те же действия. Присутствует кнопка удаления изображения.

Ниже расположены поля «краткое содержание» и «содержание поста». Первое поле – блок <textarea>, как и заголовок. Второе поле – div, отмеченный специальным id, по которому плагин ckeditor понимает, где создать поле ввода. Плагин генерирует HTML разметку, а браузер ее обрабатывает и сразу выводит на экран, из-за чего можно стилизовать различные элементы с помощью CSS.

Внизу страницы располагается селектор категорий и кнопка отправки формы. При отправке формы страница не перезагружается и не перенаправляет пользователя на другую страницу, вместо этого с помощью JavaScript создается специальный объект FormData, в который помещаются все поля формы, а также содержание тела поста, после чего этот объект отправляется на сервер с помощью axios. Ответ от сервера обрабатывается и в зависимости от результата пользователю либо поступает сообщение об ошибке, либо, в случае успеха, происходит переадресация на главную страницу. Это позволяет уменьшить задержку отправки данных и не потерять данные при переходе на другую страницу.

Для просмотров была создана отдельная таблица в базе данных. Эта таблица состоит из 4х полей:

* user\_id
* post\_id
* created\_at
* updated\_at

Первые два отвечают за связь пользователя и поста, который он просмотрел, остальные 2 для отслеживания временного отрезка. Когда пользователь запрашивает ту или иную страницу, id этой страницы добавляется в эту таблицу с пометкой, когда просмотр был совершен. Основываясь на этой таблице в дальнейшем, можно будет составить график активности просмотра и улучшать систему рекомендаций, основываясь на тематике просматриваемых постов. Для получения количества просмотров используется специальный метод модели отношений beongsToMany, которая подготавливает SQL join запрос, который можно дополнить и выбрать с помощью модификаторов запроса. Подсчитав количество пользователей в таблице, связанных с текущим постом, можно получить количество просмотров поста.

Для хранения «лайков» была создана аналогичная таблица. При нажатии на рисунок сердца браузер отправляет запрос на сервер с целью добавить новую запись в базу или удалить существующую. Во время подготовки страницы все посты получают бинарное состояние лайка, что позволяет заранее пометить нужные посты. «Лайки» скрываются, если пользователь не авторизован.

Избранное позволяет отложить просмотр поста, либо сохранить его в коллекцию. Механизм работы избранного идентичен работе лайков. Кнопка избранного скрывается если пользователь не авторизован.

Рассмотрим алгоритм получения постов из базы [9], [10], [11]. Изначальный вариант выборки информации о посте из базы данных предполагал несколько запросов в базу для каждого поста – выборка информации обо всех постах с ограничением их количества, далее для каждого поста отдельно производились запросы для получения состояния «в избранном» и «понравилось», а также запрос количества просмотров. Для сокращения количества запросов в базу данные запросы были объединены в один, посредством соединения нескольких таблиц, фильтрации и группировки полей:

protected static function booted()  
{  
 if (Auth::*check*()) { *// проверка на то, зарегистрирован ли пользователь* $user\_id = Auth::*user*()->id;  
 static::*addGlobalScope*(fn($query) => $query  
 ->orderBy('posts.created\_at', 'desc')  
 ->leftJoin('post\_likes as pl', function ($query) use ($user\_id)  
 { *// присоединение таблицы post\_likes* return $query->on('pl.user\_id', '=', DB::*raw*($user\_id))  
 ->on('pl.post\_id', '=', 'posts.id');  
 })  
 ->leftJoin('favorite as fv', function ($query) use ($user\_id)  
 { *// присоединение таблицы favorite* return $query->on('fv.user\_id', '=', DB::*raw*($user\_id))  
 ->on('fv.post\_id', '=', 'posts.id');  
 })  
 ->leftJoin('views as vw', 'vw.post\_id', '=', 'posts.id')  
 *// присоединение таблицы views* ->groupBy('posts.id', 'pl.post\_id', 'fv.post\_id')  
 *// группировка по id поста* ->select([  
 'posts.\*',  
 DB::*raw*('COUNT(pl.post\_id) as liked'),  
 DB::*raw*('COUNT(fv.post\_id) as in\_favorite'   
 DB::*raw*('COUNT(vw.\*) as views')   
 ])  
 );  
 } else  
 static::*addGlobalScope*(fn($query) => $query  
 ->orderBy('created\_at', 'desc')  
 ->leftJoin('views as vw', 'vw.post\_id', '=', 'posts.id')  
 ->groupBy('posts.id')  
 ->select(['posts.\*', DB::*raw*('COUNT(vw.\*) as views')]));  
}

Для отображения состояния лайка и избранного нужно лишь бинарное значение, а для количества просмотров – число. Исходя из этого были выбраны такие условия группировки:

* Для просмотров достаточно группировки по id поста
* Для лайков нужно указать дополнительно условие наличия записи, соответствующей конкретному посту и текущему пользователю.
* Для избранного аналогично лайкам.

Рекомендации

На сайте реализована система личных рекомендаций для пользователей. Эта система основывается на том, на какие посты пользователь поставил лайк, на категории этих постов, чтобы в дальнейшем рекомендовать посты из этих категорий. Для реализации этой системы проводится запрос к базе данных на извлечение всех категорий постов, которые были отмечены пользователем:

Category::*join*('category\_post', 'categories.id', '=', 'category\_post.category\_id')  
 ->join('posts', 'category\_post.post\_id', '=', 'posts.id')  
 ->join('post\_likes', 'category\_post.post\_id', '=', 'post\_likes.post\_id')  
 ->groupBy('categories.id')  
 ->orderBy('count', 'desc')  
 ->get(['categories.\*', DB::*raw*('count(categories.id) as count')]);

После этого вычисляется, сколько всего постов было выбрано, это число будет взято за 100%. Далее вычисляем коэффициент интереса, разделив общее количество постов в выборке на количество постов определенной категории. Этот коэффициент умножается на количество выбираемых постов и в итоге получается количество постов, нужное для выборки из определенной категории. Из каждой категории выбирается вычисленное количество постов, после чего эти посты помещаются в объект коллекции, а затем в объект пагинатора со значением общего количества постов. При переходе на новую страницу пагинатор сам передвинет offset на нужную позицию

$total = *array\_sum*(*array\_map*(fn ($category) => $category['count'],  
 $recs->toArray()));  
$posts = new Collection();  
foreach ($recs as $rec) {  
 $k = $rec->count / $total;  
 $posts = $posts->merge($rec->posts()  
 ->paginate(*round*($limit \* $k + 1, mode: *PHP\_ROUND\_HALF\_DOWN*)));  
}  
$posts = new LengthAwarePaginator($posts, Post::*all*()->count(), 15);

**Тестирование**

Для проведения тестирования были сгенерированы[23] данные для постов включая превью изображения с помощью библиотеки Faker, встроенной в Laravel. В ходе тестирования было проверено:

* Соответствие верстки постов дизайн-макету сайта
* Проверка обработки больших чисел в компонентах
* Проверка пагинации
* Проверка рекомендаций
* Проверка мобильной версии
* Проверка валидации данных
* Проверка на переполнение
* Проверка на возможность проведения SQL инъекций
* Проверка возможности просмотреть страницы, доступные только для авторизованных пользователей, не авторизованными пользователями
* Проверка мобильной версии на удобство использования

В ходе проверки были выявлены и устранены все найденные ошибки и неточности.

* 1. Руководство пользователя

При первом заходе на веб сайт отображается главная страница (рисунок 4), на которой пользователю предоставляется возможность:

* просмотреть список случайных записей,
* найти интересующую запись с помощью поиска,
* отфильтровать записи по категории,
* просмотреть последние добавленные новости в блоке «Актуальное»

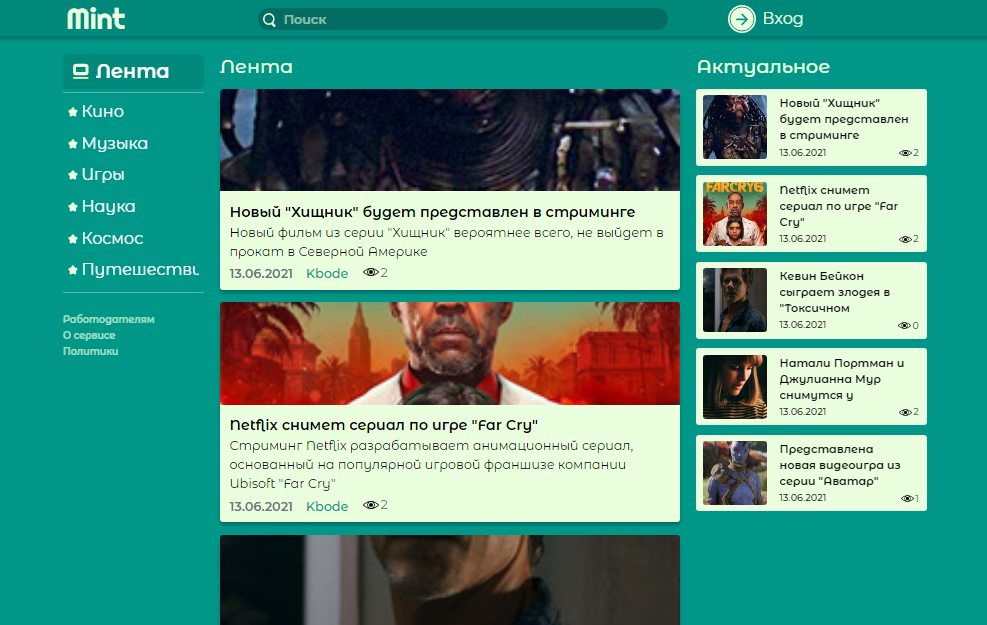


Рисунок 4 - Главная страница сайта

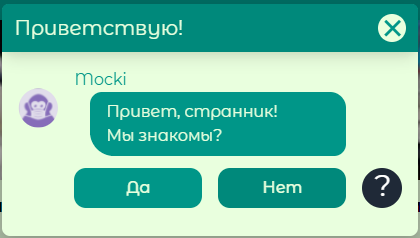
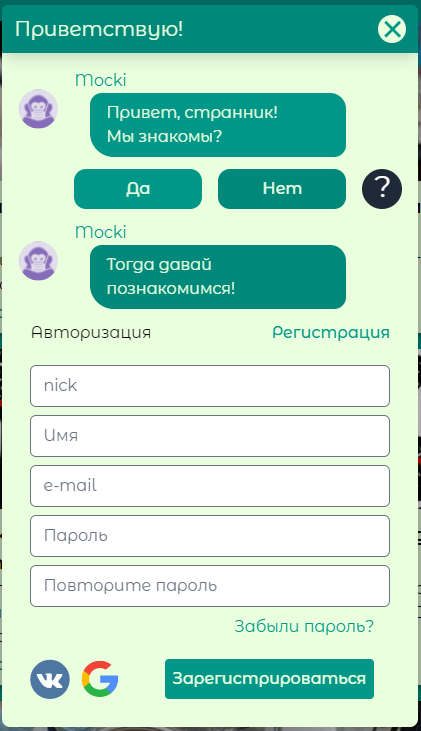
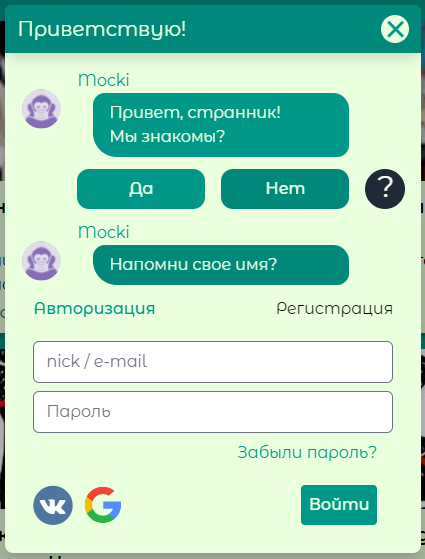
Пользователь имеет возможность зарегистрироваться на сайте. Для этого нужно открыть меню вход, расположенное в шапке сайта справа. После нажатия откроется окно pop-up, в котором будет диалоговое окно с маскотом сайта – Mocki, который задаст вопрос в виде текстового сообщения на экране с текстом «Привет, странник! Мы знакомы?», на который будет два варианта ответа – «да» и «нет». В зависимости от выбранного ответа будет открыто окно авторизации или регистрации соответственно. В случае если пользователь ошибся с выбором, то у него будет два пути смены вида формы – повторно ответить на вопрос, либо явно нажать на одну из двух кнопок – «авторизация» и «регистрация». Если была выбрана форма[22] регистрации, то появятся поля для регистрации нового пользователя, иначе останутся лишь поля ввода логина и пароля. Для авторизации подойдет как логин, так и почта. Оформление представленно на рисунке 5.

Рисунок 5 - Диалоговое окно (а), форма авторизации (б) и регистрации (в)

б)

в)

а)

После регистрации у пользователя появляются дополнительные возможности и меню (рисунок 6). В шапке появляется две иконки – первая (звезда) показывает посты, которые были добавлены в избранное. Вторая иконка открывает меню пользователя. В зависимости от роли пользователя пункты меню могут различаться.

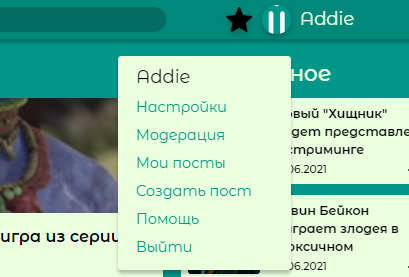
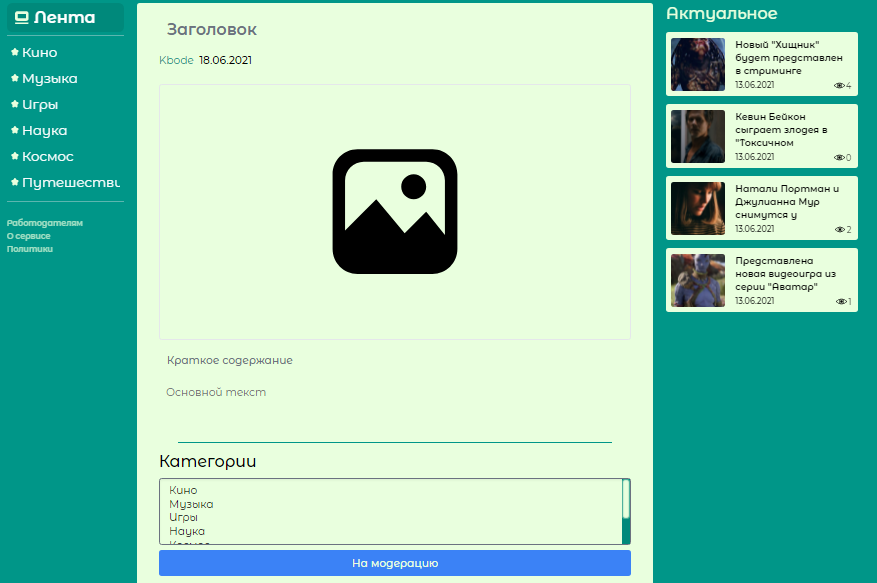
Пользователь после регистрации получает возможность добавить пост. Для этого нужно перейти на страницу создания поста (рисунок 7), заполнить все поля и нажать на кнопку «на модерацию» для дальнейшей модерации. После утверждения записи модератором она появляется в ленте новостей.

Рисунок 6 - Меню пользователя

Рисунок 7 - Страница создания поста

После создания поста его можно найти на странице постов пользователя (рисунок 8). Перейти на эту страницу можно через меню пользователя. В списке постов можно быстро просмотреть свои посты, а также их статус. Есть возможность удалить, отредактировать и просмотреть пост с помощью соответствующих иконок.

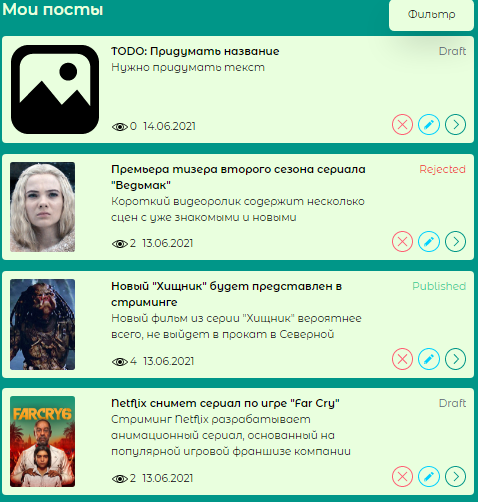


Рисунок 8 - Страница «мои посты»

Авторизованные пользователи имеют личный кабинет (рисунок 9), где можно изменить пароль и аватар. Для изменения пароля нужно заполнить соответствующие поля в форме. Для изменения аватара нужно нажать на старый аватар, выбрать новое изображение из файловой системы, после чего отобразится новый аватар. Для сохранения результата требуется нажать на кнопку сохранить под именем пользователя.

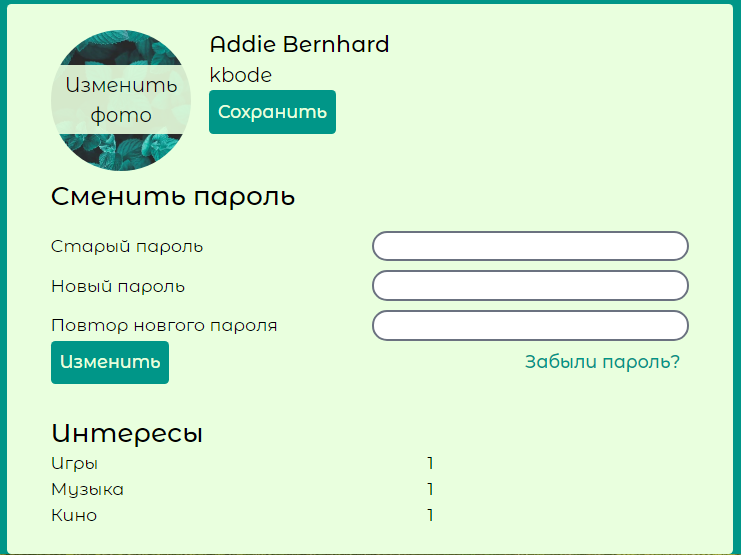


Рисунок 9 - Страница настроек пользователя

# **ЗАКЛЮЧЕНИЕ**

При выполнении ВКР была создана информационная система, которая позволяет просматривать, и создавать информационные посты с новостями. Были выполнены следующие задачи:

* Проанализирована предметная тема;
* Создан план разработки;
* Изучены инструменты разработки;
* Разработан дизайн-концепт сайта;
* Разработана база данных и информационные модели;
* Запрограммированы логические скрипты приложения;
* Сайт был наполнен контентом;
* Сайт был протестирован.

При выполнении ВКР были получены новые навыки в сфере разработки приложений, создания дизайна, проектирования баз данных. Работа является законченным программным продуктом. Были соблюдены основные нормы и правила, применяемые при создания веб приложений. Приложение спроектировано с возможностью дальнейшей поддержки и улучшения.

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

* 1. ГОСТ Р 2.105-2019 «ЕДИНАЯ СИСТЕМА КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ. ОБЩИЕ ТРЕБОВАНИЯ К ТЕКСТОВЫМ ДОКУМЕНТАМ» [Текст] - 01.02.2021 – 40с.
  2. ГОСТ 7.0-99. Система стандартов по информации, библиотечному и издательскому делу. Информационно-библиотечная деятельность, библиография. Термины и определения - 01.07.2000г – 28с .
  3. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом - 01.01.1980 – 12с.
  4. ГОСТ 24.301-80 «Система технической документации на АСУ. Общие требования к выполнению текстовых документов» - 01.01.1981 – 7с.
  5. Википедия – определение сайта [Электронный ресурс] - https://ru.wikipedia.org/wiki/Сайт;
  6. Википедия – новостной сайт [Электронный ресурс] - https://ru.wikipedia.org/wiki/ Новостной\_сайт\_(сайт\_новостей)
  7. ОСТ 29.115-88. «Оригиналы авторские и текстовые издательские. Общие технические требования» - 01.01.89 – 33с.
  8. 7.0.104-2019. Система стандартов по информации, библиотечному и издательскому делу (СИБИД). Библиотечно-информационные услуги научной библиотеки. Виды, формы и режимы предоставления – 01.06.2019.
  9. Грофф Джеймс Р., Вайнберг Пол Н. «SQL. Полное руководство» [текст] – 2018г – 960с.
  10. Фримен Эрик, Робсон Элизабет «Head First. Паттерны проектирования. Обновленное юбилейное издание» – 2018г – 656.
  11. Зандстра Мэт - PHP. Объекты, шаблоны и методики программирования – 2019 - 736c.
  12. Иванова Г.С. ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ (ДЛЯ БАКАЛАВРОВ). – М.: , 2016. – 336 с.
  13. В.В. Дунаев. (Х)HTML, скрипты и стили. Самое необходимое. – СПб.: БХВ-Петербург, 2012. – 496 с.
  14. Симпсон Кайл «Асинхронная обработка и оптимизация» - 2019 – 352с.
  15. Документация к PHP фреймворку Laravel версии 8.Х [Электронный ресурс] – https://laravel.com/api/8.x/
  16. Документация к CSS библиотеке TaiwindCSS [Электронный ресурс] – https://tailwindcss.com/docs
  17. Документация к фреймворку laravel-livewire 2.x [Электронный ресурс] – https://laravel-livewire.com/docs/2.x/quickstart
  18. Документация к JavaScript библиотеке AlpineJS 2.8.0 [Электронный ресурс] – https://github.com/alpinejs/alpine (ru, en)
  19. Работа с Blade компонентами [Электронный ресурс] –https://laravel.com/docs/8.x/blade
  20. Правила построения баз данных для создания отношений между моделями [Электронный ресурс] –https://laravel.com/docs/8.x/eloquent-relationships
  21. Создание REST API в Laravel [Электронный ресурс] –https://laravel.com/docs/8.x/eloquent-resources
  22. CSRF Защита при произведении межсайтового запроса [Электронный ресурс] – https://laravel.com/docs/8.x/csrf
  23. Паттерн фабрики, создание тестовых данных в базе данных [Электронный ресурс] – https://laravel.com/docs/8.x/database-testing
  24. Создание миграций базы данных, работа с типами данных [Электронный ресурс] – https://laravel.com/docs/8.x/migrations
  25. Установка CKEditor5 [Электронный ресурс] – https://ckeditor.com/docs/ckeditor5/latest/builds/guides/integration/installation.html
  26. Базовые функции CKEditor5 [Электронный ресурс] – https://ckeditor.com/docs/ckeditor5/latest/builds/guides/integration/basic-api.html
  27. Конфигурирование CKEditor5 [Электронный ресурс] – https://ckeditor.com/docs/ckeditor5/latest/builds/guides/integration/configuration.html
  28. Основы Material Design [Электронный ресурс] – https://material.io/design/introduction
  29. Валидация данных [Электронный ресурс] – https://laravel.com/docs/8.x/validation
  30. Современный учебник JavaScript [Электронный ресурс] - https://learn.javascript.ru
  31. Документация к языку PHP 8.0 [Электронный ресурс] - https://www.php.net/docs.php
  32. Электронный учебник по HTML5 [Электронный ресурс] - https://html5book.ru/html-html5

Приложение A – ER-диаграмма

