|  |  |
| --- | --- |
| Изображение выглядит как символ  Автоматически созданное описание | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования  «Новгородский государственный университет имени Ярослава Мудрого»  **ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ** |

# Разработка сайта автомобильных новостей “Alpha-News”

Курсовой проект по учебной дисциплине

“Разработка кода информационных систем”

По специальность 09.02.07 “Информационные системы и программирование”

ПТК.КП 2993 12.000ПЗ

|  |
| --- |
| Студент группы 2994  \_\_\_\_\_\_\_\_/ М. В. Костив /  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель  \_\_\_\_\_\_\_\_\_/ В. Р. Ильин / |

# Содержание

Оглавление

[Разработка сайта автомобильных новостей “Alpha-News” 1](#_Toc164899663)

[Содержание 2](#_Toc164899664)

[Аннотация 4](#_Toc164899665)

[Введение 5](#_Toc164899666)

[Общая (теоретическая) часть. 7](#_Toc164899667)

[Постановка задач (ТЗ) 7](#_Toc164899668)

[Характеристики бизнес-процессов. 7](#_Toc164899669)

[Технико-математическое описание задачи 8](#_Toc164899670)

[Требования к программе 10](#_Toc164899671)

[Обоснование проектных решений 10](#_Toc164899672)

[Обоснование выбора языков программирования 12](#_Toc164899673)

[Обоснование выбора среды программирования 13](#_Toc164899674)

[Информационное обеспечение 14](#_Toc164899675)

[Обзор и анализ существующих программных систем 14](#_Toc164899676)

[Раздел 2. Практическая составляющая работы 15](#_Toc164899677)

[Обоснование и описание выбора состава технических и программных средств 15](#_Toc164899678)

[Практическая реализация алгоритмов 17](#_Toc164899679)

[Тестирование 30](#_Toc164899680)

[Заключение 34](#_Toc164899681)

[Список литературы 35](#_Toc164899682)

[Приложение A. 36](#_Toc164899683)

[Idef0-диаграмма 36](#_Toc164899684)

# Аннотация

Данный проект представляет собой разработку сайта автомобильных новостей, с возможностью продажи услуг по продвижению проектов. На данном сайте будут собраны свежие новости с различных источников. Плюсом будет то, что сами пользователи смогут опубликовывать свои новости при покупке одного из пакетов услуг.

# Введение

Данный сайт будет разработан для максимально широкой аудитории, для людей которым интересен мир автомобилей. Данный сайт нацелен на извлечение прибыли из информационного поля путем продажи рекламных услуг.

На большинстве сайтов конкурентов неудобные элементы управления и устаревший дизайн. Визуальное восприятие в конкретном типе сайтов очень важно, так как по большей части, обычный пользователь заходит на подобные ресурсы с целью получить свежую информацию и расслабиться прочтением интересных новостей, поэтому над дизайном и интерфейсом была проведена кропотливая работа. Данный проект будет выделяться на фоне конкурентов своим минимализмом и простоте эксплуатации. Основная концепция данного сайта – позволить пользователям опубликовывать различные новости, связанные с автомобильным миром. Для привлечения компаний, с которыми в дальнейшем планируется сотрудничество на взаимовыгодных условиях мы планируем привлечь наибольшее количество пользователей.

У новостей будет счетчик просмотров и возможность оставлять комментарии, что – бы компании видели, что у нас есть живая аудитория, которые просматривают наши новости и мы сможем принести пользу сторонним проектам. Очень важно удержание аудитории, постоянное количество пользователей, которые регулярно просматривают и комментируют публикации на самом сайте и размещения пользователей, рекламных записей. Для того, чтобы удержать пользователя, мы предлагаем широкий спектр новостей из мира автомобилей, собранных из различных ресурсов. Плюсом будет наличие колонки с мероприятиями, которые можно будет посетить вживую, в дальнейшем опубликовать фотографии с мероприятия и оставить свой отзыв. Про блок с событиями, наш сайт предлагает услугу продажи билетов на ваши мероприятия. Купить билет можно будет напрямую через сайт. Плюс для организаторов будет большой поток людей на мероприятии, для сайта же, будет выбран определенный процент с продаж билетов через меню сайта.

Цель: разработать сайт автомобильных новостей с интуитивно понятным интерфейсом и приятным глазу дизайном, что бы пользователям было максимально комфортно длительно проводить время за прочтением новостей. Привлечь наибольшее количество аудитории, для извлечения из этого прибыли, путем продажи услуг по продвижению сторонним компаниям, получением процента с продажи билетов на события через сайт.

Задачи:

* Создание приятного глазу дизайна, максимально минимизирующего нагрузку на глаза.
* Разработка интуитивно понятного интерфейса для пользователя
* Разработать функционал для более удобного прочтения новостей
* Разработать функционал личного кабинета, для того, чтобы отслеживать активность пользователей на сайте, удержание аудитории.
* Оптимизировать сайт, что – бы он мог спокойно функционировать при большом количестве пользователей.
* Выделиться на фоне конкурентов удобным интерфейсом, приятным дизайном.
* Привлечь наибольшее количество аудитории для извлечения прибыли в дальнейшем.
* Привлечь организации, бизнес проекты, готовые приобретать наши услуги.

# Общая (теоретическая) часть.

# Постановка задач (ТЗ)

На данном этапе обучения, требуется разработать первичную версию сайта автомобильных новостей “Alpha-News” с рабочим функционалом в меру собственных знаний. Для того, что бы в дальнейшем ввести сайт в эксплуатацию.

По функционалу данный сайт должен обладать следующими особенностями: Полностью функционирующая система регистрации, работающая система модальных окон для прочтения новостей. В дальнейшем планируются разработка функционала добавления новостей в избранное, отслеживание количества просмотров на новости, система оставления комментариев различными пользователями.

# Характеристики бизнес-процессов.

Ресурсами данного бизнес процесса являются. 1 персональный компьютер, интернет. Разработка будет осуществляться через интерпретатор кода Visual Studio Code. При помощи языка программирования Java Script, языка текстовой разметки HTML и каскадных таблиц стилей (CSS).

Этапы разработки:

1. Разработка дизайна в Figma.
2. Написание верстки.
3. Разработка функционала

Утилиты:

1. Figma
2. Visual Studio Code

К основным проблемам можно отнести:

1. Ограниченные знания в области программирования

При успешной реализации данного бизнес-процесса мы получим конечный продукт в виде функционирующего новостного сайта Alpha-News.

При реализации данного бизнес-процесса могут возникнуть следующие проблемы:

1. Ограниченные возможности серверов, невозможность поддерживать работоспособность сайта при одновременном нахождении на нем множества пользователей.
2. Первое время, сайт может не приносить никакой прибыли, нужно некоторое время на “раскрутку” сайта в информационном поле путем рекламы на различных интернет ресурсах.
3. Отсутствие адаптивности сайта под мобильные устройства. Основное внимание сейчас заточено на разработке функционала сайта и улучшение дизайна, в будущем адаптивность будет присутствовать.
4. Частичное несоответствие дизайна в Figma и конечного продукта. Некоторые моменты могут быть изменены при тестировании продукта на различных устройствах. Сейчас очень важно, что бы сайт хорошо выглядел на совершенно различных устройствах.

# Технико-математическое описание задачи

При введении в эксплуатацию должен быть реализован весь вышеперечисленный функционал, например: система регистрации и авторизации, работающий личный кабинет с информацией о избранных новостях и персональными данными аккаунта, модальные окна для удобного просмотра новостей, система добавления новостей на сайт, размещение отзывов к публикациям, система добавления новостей в избранное и их отображения в личном кабинете. Должен быть реализован интуитивно понятный интерфейс для пользователя, грамотно выбранная цветовая палитра для длительного чтения новостей.

Вся работа по реализации данного проекта будет проходить в интерпретаторе кода Visual Studio Code при помощи языка Java Script без применения сторонних фреймворков на данном этапе разработки. В дальнейшем планируется подключение БД к проекту при помощи PHP и переведение сайта на tailwindcss, это значительно упростит дальнейшую работу с версткой сайта и взаимодействий языка Java Script и CSS. Это позволит значительно сократить CSS-код.

Весь функционал будет написан при помощи языка Java Script.

Стили и анимации оформлены через keyframes при помощи языка CSS.

Запуск большинства анимаций реализован через Java Script, :hover-анимации (анимация при наведении на определенный элемент) полностью реализованы через CSS.

Регистрация на данном этапе будет реализована без применения сторонних баз данных, как альтернатива будет использовано локальное хранилище (localStorage). Взаимодействие с локальным хранилищем реализовано при помощи Java Script с использованием методов преобразования данных JSON (parse, stringify) и методов localStorage (getItem(item\_name), setItem(‘key’, value)).

Открытие модальных окон реализовано при помощи CSS в паре с языком программирования Java Script. Изначально в HTML создана секция со скрытым через каскадные таблицы стилей (CSS) содержимым (display: none), при нажатии на нужную нам новость через JS достается ID(уникальный идентификатор) новости, которую нужно открыть и через нужный нам ID изменяются стили конкретного модального окна, мы видим нужное нам содержимое

Обработчики ошибок в регистрации будут реализованы через HTML-тег <dialog></dialog> с атрибутом open=””. Изначально dialog не имеет никаких атрибутов, при возникновении ошибки в регистрации, через JS мы задаем в dialog атрибут open=”” и мы даем пользователю знать, что в данном поле информация заполнена неправильно. Анимации про проявлении диалогового окна будут инициализированы через JS, методом взаимодействия JS с CSS (document.getElementById(idElem).style.animation = “anim\_name 1s ease-in-out”).

# Требования к программе

Работа основополагающего функционала сайта – модальные окна и система регистрации/авторизации.

Разработанный интуитивно понятный пользователю интерфейс с использованием Figma.

Разработанный приятный для длительного нахождения на странице дизайн с использованием методов UI/UX.

Анимации на сайте должны начинаться и завершаться незаметно для пользователя, при воспроизведении анимации не должно быть никаких визуальных артефактов.

Совместимость с различными браузерами.

Сайт на данном этапе должен быть выполнен с использованием языка программирования Java Script, языка текстовой разметки HTML и стилизации при помощи CSS.

# Обоснование проектных решений

Обоснование выбора модальных окон для прочтения новостей.

Модальные окна как техническое решение намного упрощают и сокращают время на написание кода, были еще варианты для исполнения данной функции, такие как:

1. Создание отдельных страниц для каждой новости.

Данный вариант был слишком нерентабелен для масштабов данного проекта, пришлось бы создавать более 40 страниц с новостями, и для каждой отдельной страницы прописывать свой HTML код. Сложнее было бы в дальнейшем взаимодействовать и доставать информацию о допустим, просмотрах на новости, оставленных комментариев. В связи с недостаточным количеством знаний, невозможность реализации функции добавления новостей. Т.к при открытии каждой новости приходилось бы создавать новую html страницу. Возможно, это реализуемо при помощи языка PHP, в дальнейшем будут увеличиваться масштабы сайта, будут применены большие объемы информации, придется переносить все на базу данных.

1. Создание одной страницы для всех новостей.

Эта идея реализации была отвергнута по причине увеличения объема JS-кода. Т.к нужно было бы вызывать функцию добавления данных, присваивание своих уникальных идентификаторов каждой новости, ссылок на изображения. Опять же, все упирается в объем знаний в области PHP. При наличии базы данных, это вполне реализуемая идея, но со своими трудностями.

По данным причинам, было выбрано самое оптимальное решение для реализации конкретной функции – модальные окна. ID заранее заданы, нужные изображения и текстовая информация. Через JS, при нажатии на конкретную новость, которую вы хотите открыть, вызывается функция, которая через обработчик событий на странице, получает ID данного элемента, и по этому идентификатору открывает нужное модальное окно с информацией. Данное решение на данном этапе является самым оптимальным, так как:

1. Минимальный объем JS кода
2. Простота в написании
3. Экономия времени
4. Удобство редактирования при надобности
5. Удобство добавления новых модальных окон для новостей, т.к есть уже заранее заготовленные контейнеры, все что требуется – задать нужный ID.

# Обоснование выбора языков программирования

HTML – язык текстовой разметки, позволяет web-разработчику работать с большим количеством текстовой информации, редактировать и частично стилизовать. HTML является базовым языком верстки страницы, т.к все блоки, секции с которыми взаимодействуют CSS и JS написаны изначально в HTML. Вся видимая информация на странице написана в HTML формате. Позволяет разработчику писать текст и различные блоки входной информации напрямую на страницу, создание секций, позволяет задавать уникальные идентификаторы отдельным элементам, тексту или целым блокам информации.

CSS – каскадные таблицы стилей, которые через отдельный файл, или же напрямую в HTML файле через тэг <style></style> . CSS также является одним из основных инструментов web-разработчика, который позволяет разработчику присваивать стили целым блокам или отдельным элементам. Значительно упрощает процедуру разработки web-страницы, позволяет задавать размеры изображений, размер текста, толщину текста, размер целых секций, блоков. Позволяет изменять цвета заднего фона на странице и различных элементов. Также, благодаря CSS возможно создавать адаптивность сайта под различные разрешения экранов.

Java Script – язык программирования, который позволяет написать различный функционал для сайта. Язык очень широкого применения, позволяет взаимодействовать с CSS и HTML посредством функций. JS обладает множеством методов для компиляции информации, изменения стилей, получение входных данных напрямую со страницы. Позволяет создавать обработчики событий, допустим, при нажатии на кнопку, при пролистывании страницы, при загрузке страницы. Также, позволяет взаимодействовать с локальным хранилищем браузера, вносить туда изменения и создавать новые элементы в HTML странице.

# Обоснование выбора среды программирования

Visual Studio Code – бесплатная утилита от Microsoft, которая значительно упрощает взаимодействие с различными языками программирования на одном сайте. Т.к можно устанавливать бесконечное множество дополнительных библиотек по своим потребностям. Программистам приходится проводить в среде разработки достаточно длительное время, при этом очень большое значение имеет правильная цветовая гамма, которая не будет давить на зрительные органы яркими цветами и возможностью выбирать цветовую гамму под свой вкус. В данной утилите приятная цветовая гамма, которая позволяет проводить достаточно длительное время за разработкой. Взаимодействие с программой максимально удобно, можно напрямую создавать новые файлы, максимально сокращает временные потери при написании кода.

Figma – браузерная утилита, которая позволяет разрабатывать дизайн сайта. Удобна тем, что можно создавать анимации напрямую в Figma и в последующем экспортировать код напрямую в среду разработки, частично упрощает написание CSS кода. Позволяет при помощи дополнительных библиотек самостоятельно создавать цветовую палитру через собственные алгоритмы. Вся разработка дизайна сайта, кнопок, анимаций, полей для входных данных, блоков производится именно в Figma, т.к она напрямую заточена под данную задачу. Также как альтернативу можно использовать Photoshop, но изначально, Photoshop является профессиональным инструментом для обработки фотографий. Требуется большое количество времени на освоение всех возможностей утилиты, Figma же является очень простой в освоении и эксплуатации, обладает множеством горячих клавиш, которые также сокращают время разработки дизайна.

# Информационное обеспечение

Для новостного сайта Alpha-News были выявлены следующие средства информационного обеспечения:

Хранение информации о зарегистрированных пользователях и данные о добавленных новостях будут хранится посредством локального хранилища браузера в JSON формате.

Для разработки основного функционала сайта был выбран язык программирования Java Script, без использования сторонних фреймворков и библиотек на данном этапе обучения.

Защита сайта от DDOS-атак обеспечивается браузерным протоколом HTTP. Защита от взлома аккаунта обеспечивается базой данных с зарегистрированными пользователями, в дальнейшем планируется для улучшения безопасности использовать хэширование логинов и паролей через протокол MD5.

# Обзор и анализ существующих программных систем

Autonews.ru

Сайт: https://www.autonews.ru/

Относительно давно созданный ресурс с множеством новостных статей. Но приоритетом там находятся новости одной категории. На нашем же ресурсе предлагается больше разделов с новостями из автомобильного мира: Формула 1, RDS, события, автомобилестроение. Цветовая гамма на данном ресурсе не способствуют длительному нахождению на сайте – используются слишком яркие цвета.

Преимущество обсуждаемого ресурса в том, что присутствует список наилучших автомобилей в определенной ценовой категории по рейтингу и отзывам с различных ресурсов. Большое количество технической документации к различным автомобилям.

6auto.ru

Сайт: <https://6auto.ru/>

Данный ресурс в эксплуатации очень длительное время. Это также сайт автомобильных новостей с большим количеством категорий. В связи с тем, что сайт особо не изменился с момента введения в эксплуатацию у него устаревший дизайн, но цветовая гамма в целом приятна для пользователя. За счет того, что ресурс существует достаточно давно, в нем находится огромное количество информации, которую тяжело найти на других ресурсах. Присутствует огромное количество документации к автомобилям которой не найти на других сайтах, много описания особенностей ремонта и эксплуатации. Из минусов можно подчеркнуть:

1. Устаревший дизайн
2. Часто долгая загрузка новостей
3. При эксплуатации на мониторах с меньшим разрешением не масштабируются некоторые изображения.

Плюсы:

1. Большое количество информации
2. Множество различный категорий новостей

# Раздел 2. Практическая составляющая работы

Реализация

# Обоснование и описание выбора состава технических и программных средств

Для реализации будут использоваться следующие утилиты:

1. Visual Studio Code
2. Google Chrome
3. Figma

Visual Studio Code – бесплатная утилита от Microsoft, которая значительно упрощает взаимодействие с различными языками программирования на одном сайте. Т.к можно устанавливать бесконечное множество дополнительных библиотек по своим потребностям. Программистам приходится проводить в среде разработки достаточно длительное время, при этом очень большое значение имеет правильная цветовая гамма, которая не будет давить на зрительные органы яркими цветами и возможностью выбирать цветовую гамму под свой вкус. В данной утилите приятная цветовая гамма, которая позволяет проводить достаточно длительное время за разработкой. Взаимодействие с программой максимально удобно, можно напрямую создавать новые файлы, максимально сокращает временные потери при написании кода.

Figma – браузерная утилита, которая позволяет разрабатывать дизайн сайта. Удобна тем, что можно создавать анимации напрямую в Figma и в последующем экспортировать код напрямую в среду разработки, частично упрощает написание CSS кода. Позволяет при помощи дополнительных библиотек самостоятельно создавать цветовую палитру через собственные алгоритмы. Вся разработка дизайна сайта, кнопок, анимаций, полей для входных данных, блоков производится именно в Figma, т.к она напрямую заточена под данную задачу. Также как альтернативу можно использовать Photoshop, но изначально, Photoshop является профессиональным инструментом для обработки фотографий. Требуется большое количество времени на освоение всех возможностей утилиты, Figma же является очень простой в освоении и эксплуатации, обладает множеством горячих клавиш, которые также сокращают время разработки дизайна.

Google Chrome – веб-браузер, созданный на основе браузера Chromium. В настоящее время Google Chrome остается одним из самых используемых браузеров в мире благодаря своей скорости, безопасности и обширным возможностям расширений. Браузер также активно обновляется и совершенствуется, чтобы соответствовать современным стандартам веб-разработки и потребностям пользователей.

# Практическая реализация алгоритмов

Практическая реализация модальных окон – для начала было создано множество контейнеров с одинаковыми классами, но каждым уникальным идентификатором, для того, что бы через нажатие на новость открывалась конкретное модальное окно с заранее заданной информацией. Данная система работает таким образом, что при прогрузке DOM-элементов присутствует функция, которая через цикл for в Java Script получает массив HTML элементов, в последующем каждому из этих элементов присваивается обработчик событий addEventListener. У addEventListener присутствует атрибут event, через который возможно достать информацию о том элементе, на который вы нажали и соответственно, достать id модального окна, которое нужно открыть. Изначально контейнер модального окна имеет стиль в CSS display: none, то есть элемент изначально не отображается на странице. При нажатии на новость, через event мы достаем id контейнера модального окна и при помощи Java Script мы изменяем стиль контейнера модального окна на display: block. При присваивании данного стиля, нужное нам модальное окно проявляется на странице. После того, как нужное нам модальное окно отображено, в коде присутствует скрипт, который при нажатии на клавишу escape закрывает модальное окно. Также присутствует код, который убирает прокрутку страницы вниз или вверх, при закрытии модального окна прокрутка страницы включается снова.

HTML-код модального окна:

<section class="modalAppSection" id="RDSmodal\_six">

<article class="modalContent" id="RDSmodal\_six">

<img src="images/rds/rdsNewFour.png" alt="">

<section class="modalContenAuthorAndDate">

<article><p>Дарья Бугаева, 21 мар. 2023</p></article>

<article><p>Alpha-News</p></article></section>

<section class="modalAppText"><p></p></section></article></section>

JS-код:

function modalAppOpen (id) {

document.querySelector(id).style.display = "grid"

document.querySelector(`${id}Con`).style.display = "flex"

if (!document.querySelector(id).style.display === "grid") {document.body.style.overflow = "unset"

} else {document.body.style.overflow = "hidden"}}

function closeModal (id) {

document.body.addEventListener("keydown", (event) => {

if(event.key === "Escape") {

document.querySelector(id).style.display = "none";

document.querySelector(`${id}Con`).style.display = "none";

document.body.style.overflow = "unset"}})}

const imgElementsHTML = document.body.querySelectorAll(".modalTrigger")

imgElementsHTML.forEach((el) => {

el.addEventListener("click", (event) => {

let elementId = [...event.srcElement.classList]

elementId = elementId[2]

modalAppOpen(`#${elementId}`)

closeModal(`#${elementId}`)})})

// При DOMContentLoaded //

const imgElementsHTML = document.body.querySelectorAll(".modalTrigger")

imgElementsHTML.forEach((el) => {

el.addEventListener("click", (event) => {

let elementId = [...event.srcElement.classList]

elementId = elementId[2]

modalAppOpen(`#${elementId}`)

closeModal(`#${elementId}`)})})

Практическая реализация функционала системы регистрации – система регистрации работает через форму с полями для ввода данных. Присутствует поля: логин, пароль, повторите пароль, email, номер телефона, также есть кнопка зарегистрироваться и “Уже есть аккаунт”. Также присутствует обработчик событий, реализованный через конструкцию else if. Обработчик событий посылает уведомление (тэг <dialog></dialog> в HTML) через диалоговое окно о том, что либо вы не ввели данные в указанное поле, либо ввели недостаточно данных. При успешной регистрации также будет выведено уведомление, затем добавлены данные в массив пользователей, затем отправлены в localStorage путем методов JSON.stringify (преобразование данных в JSON формат) и методов локального хранилища setItem(key, value (поле value обязательно должно быть в JSON формате)) . После регистрации нужно будет авторизоваться. В авторизации всего 2 поля: логин, пароль и кнопка войти. При вводе данных и нажатия на кнопку войти, данные из полей сравниваются с данными в массиве (путем forEach JS – метода переборки массива и условием if (условие) {если true}) с зарегистрированными пользователями. После успешной авторизации, в локальное хранилище через метод setItem() создается элемент с информацией о конкретном аккаунте пользователя, в который вы вошли и элемент auth = true. При auth = true вам доступен весь функционал сайта. Если данные совпали, вам открывается доступ в личный кабинет и полноценный функционал сайта. База данных представляет собой массив с объектами:

({

login: login.value,

password: password.value,

forgotPassword: forgotPassword.value,

email: email.value,

phone: numberPhone.value

})

HTML-код:

<a href="" id="LKbutton">Личный кабинет</a>

<section id="accordeonReg">

<dialog id="authDialogApp"></dialog>

<article id="accordeonRegContent">

<h1 id="regAccordeonH1">Регистрация</h1>

<p id="loginAuth" class="authDisplay">Ваш логин: #####</p>

<p id="emailAuth" class="authDisplay">Ваша почта: ####</p>

<button id="outAccount" class="authDisplay">Выйти</button>

<form name="regForm" id="regForm">

<dialog id="loginErrorEmpty"></dialog>

<input id="login" type="text" name="login" placeholder="Логин" minlength="4" maxlength="8"> <input type="text" name="password" placeholder="Пароль" minlength="6" maxlength="12">

<input type="text" name="forgotPassword" placeholder="Повторите пароль" minlength="6" maxlength="12">

<input type="text" name="email" placeholder="Введите почту" minlength="8" maxlength="40">

<input type="text" name="numberPhone" placeholder="Номер телефона" maxlength="12">

<button name="submit">Зарегестрироваться</button>

</form>

<form name="authForm" id="authForm">

<input id="login" type="text" name="login" placeholder="Логин" minlength="4" maxlength="8">

<input type="text" name="password" placeholder="Пароль" minlength="6" maxlength="12">

<button name="submit">Авторизоваться</button> </form>

<button id="authButton">Уже есть аккаунт?</button>

</article></section>

JS-код:

let users = []

if (localStorage.getItem("users")) {users = JSON.parse(localStorage.getItem("users"))}

document.forms.regForm.submit.addEventListener("click", (event) => {

event.preventDefault()

let login = document.forms.regForm.login

let password = document.forms.regForm.password

let forgotPassword = document.forms.regForm.forgotPassword

let email = document.forms.regForm.email

let numberPhone = document.forms.regForm.numberPhone

let errText = null

let error = null

function err() {

if (login.value.length === 0) {return errText = `Ошибка: поле "${login.placeholder}" не заполнено`

} else if (login.value.length < 4) {return errText = `Ошибка: поле "${login.placeholder}" меньше 4 символов`

} else if (password.value.length === 0) {return errText = `Ошибка: поле "${password.placeholder}" не заполнено`

} else if (password.value.length < 6) {return errText = `Ошибка: поле "${password.placeholder}" меньше 6 символов`

} else if (forgotPassword.value.length === 0) {return errText = `Ошибка: поле "${forgotPassword.placeholder}" не заполнено`

} else if (forgotPassword.value.length < 6) {return errText = `Ошибка: поле "${forgotPassword.placeholder}" меньше 6 символов`

} else if (email.value.length === 0) {return errText = `Ошибка: поле "${email.placeholder}" не заполнено`

} else if (email.value.length < 8) {return errText = `Ошибка: поле "${email.placeholder}" меньше 8 символов`

} else if (numberPhone.value.length === 0) {return errText = `Ошибка: поле "${numberPhone.placeholder}" не заполнено`

} else if (numberPhone.value.length < 10) {return errText = `Ошибка: поле "${numberPhone.placeholder}" меньше 10 символов`

} else {error = false}

}

err()

if (login.value.length < 4 || password.value.length < 6 || forgotPassword.value.length < 6 || email.value.length < 8 || numberPhone.value.length < 10) {

document.getElementById('loginErrorEmpty').textContent = errText

if (!document.getElementById('loginErrorEmpty').hasAttribute("open")) {

return document.getElementById('loginErrorEmpty').toggleAttribute("open")}} else { document.getElementById('loginErrorEmpty').removeAttribute("open")} if (error === false) {

users.push({

login: login.value,

password: password.value,

forgotPassword: forgotPassword.value,

email: email.value,

phone: numberPhone.value})

localStorage.setItem("users", JSON.stringify(users))

errText = "Регистрация произведена успешно" document.getElementById('loginErrorEmpty').style.background = "green"

document.getElementById('loginErrorEmpty').textContent = errText document.getElementById('loginErrorEmpty').toggleAttribute("open")

setTimeout(() => {document.getElementById('loginErrorEmpty').removeAttribute("open")}, 2000)

} else {return}

})

Реализация функционала добавления новостей – присутствует секция с новостями, в которую, при том условии, что вы вошли в личный кабинет, вам будет доступна функция “Предложить новость”. Нажимая на данную кнопку, скрипт проверяет наличие элемента auth в локальном хранилище, если он существует, если элемент отсутствует, то основной код не выполнится, будет выведено диалоговое окно о том, что вы не вошли в аккаунт открывается модальное окно с вводом данных, туда через input-поля можно вставить ссылку на изображение, автора, заголовок и подзаголовок + текст самой новости в модальном окне. Когда новость будет добавлена, через JS присваивается уникальный id для самого окна и уникальный id для модального окна. Далее скрипт добавляет обработчик событий на новость, что бы подключилась система добавления новостей, после все данные, которые нужны для добавления новости отправляются в локальное хранилище браузера. При перезагрузке страницы элемент создается заново.

HTML-код

<button id="addNewsAutocraftOpen" class="buttonTrueBackground addNewsButton">Предложить новость</button>

<dialog id="dialogAddNews">Вы не вошли в аккаунт!</dialog>

<section id="modalAppAddNews" class="modal">

<article id="modalAppAddNewsContent">

<h1>Добавить новости</h1>

<form name="formAddNewsAutoCraft" id="formAddNews">

<input name="src" id="srcImage" placeholder="Введите ссылку на изображение">

<input name="author" placeholder="Автор">

<input name="h1"placeholder="Заголовок">

<input name="h2" placeholder="Подзаголовок">

<textarea name="text" cols="30" rows="10"></textarea>

<button id="addNewsAutoCraft">Предложить новость</button>

</form>

</article>

JS-код:

document.body.addEventListener('keydown', (ev) => {

ev.key === 'Escape' ? document.getElementById('modalAppAddNews').style.display = 'none' : false

document.getElementById('modalAppAddNews').style.display === 'block' ? document.body.style.overflow = "hidden" : document.body.style.overflow = "unset"

})

document.getElementById('addNewsAutocraftOpen').addEventListener('click', (ev) => {

if (JSON.parse(localStorage.getItem('auth')) === true) {

document.getElementById('modalAppAddNews').style.display = 'block'

document.getElementById('modalAppAddNews').style.display === 'block' ? document.body.style.overflow = "hidden" : document.body.style.overflow = "unset"

} else {

document.getElementById('dialogAddNews').toggleAttribute('open')

setTimeout(() => {document.getElementById('dialogAddNews').style.animation = 'addNewdialogAppClose .3s infinite'}, 2000)

setTimeout(() => {document.getElementById('dialogAddNews').toggleAttribute('open');document.getElementById('dialogAddNews').style.animation = 'addNewDialogOpen .2s ease-in-out'}, 2200)

}

document.getElementById('addNewsAutoCraft').addEventListener('click', (event) => {

event.preventDefault()

let srcImage = document.forms.formAddNewsAutoCraft.src.value

let text = document.forms.formAddNewsAutoCraft.text.value

let author = document.forms.formAddNewsAutoCraft.author.value

let h1 = document.forms.formAddNewsAutoCraft.h1.value

let h2 = document.forms.formAddNewsAutoCraft.h2.value

let secAdd = document.getElementById('autocrSec')

console.log(secAdd)

let date = new Date

console.log(date.getFullYear())

setTimeout(() => {

secAdd.insertAdjacentHTML('beforebegin', `<article class="newBase">

<img src="${srcImage}" class="imgBaseNew modalTrigger autoCraftModal\_${addedNews.length}">

<section class="modalAppSection" id="autoCraftModal\_${addedNews.length}">

<article class="modalContent" id="autoCraftModal\_${addedNews.length}Con">

<img src="${srcImage}" alt="">

<section class="modalContenAuthorAndDate">

<article><p>${author}, ${date.getDate()}.${date.getUTCMonth()}.${date.getFullYear()}</p></article>

<article><p>Alpha-News</p></article>

</section>

<section class="modalAppText">

<h1>${h1}</h1>

<p>${text}</p>

</section>

</article>

</section>

<span>

<p class="nameNew">Автомобилестроение</p>

<p class="date">${date.getDate()}.${date.getUTCMonth()}.${date.getFullYear()}</p>

</span>

<h3>${h1}</h3>

<h4>${h2}</h4>

</article>`)

}, 2000)

addedNews.push({

id: addedNews.length,

src: srcImage,

textContent: text,

authorNew: author,

zag: h1,

podZag: h2,

secAdd: secAdd

})

const imgElementsHTML = document.body.querySelectorAll(".modalTrigger")

imgElementsHTML.forEach((el) => {

el.addEventListener("click", (event) => {

let elementId = [...event.srcElement.classList]

elementId = elementId[2]

modalAppOpen(`#${elementId}`)

closeModal(`#${elementId}`)

})

})

localStorage.setItem('newsAdded', JSON.stringify(addedNews))

})

Открытие диалоговых окон – диалоговые окна создаются тэгом html <dialog></dialog> и работают следующим образом. Через JS-код, допустим, при нажатии кнопки или при выполнении определенного условия, диалоговому окну присваивается атрибут open=”” и диалоговое окно открывается.

HTML-код:

<dialog id="dialogAddNews">Вы не вошли в аккаунт!</dialog>

JS-код:

document.getElementById('dialogAddNews').toggleAttribute('open')

Реализация интерфейса

Интерфейс был реализован с использование стандартов UI/UX дизайна через веб-версию утилиты Figma. На сайте была выбрана следующая цветовая гамма.

Цвета заднего фона:

1. #333333
2. #272727

Цвета заднего фона были выбраны специально таким образом, что бы пользователю было комфортно находится на сайте продолжительное время. Темные цвета меньше давят на глаза.

Основные цвета сайта:

1. #91705F
2. #E76D14

Цвет под номером один является основным, цвет под вторым номером используется для некоторых блоков сайта для привлечения внимания. Изначально, основным цветом сайта был цвет под номером 2 (ярко-оранжевый), но мною было принято решение, что этот цвет слишком агрессивный для данной тематики сайта как основной. Поэтому как основной цвет был выбран более мягкий – цвет под номером 1 (светло-коричневый), он является более приятным глазу и создает ощущение уюта.

Все блоки новостей являются строго структурированными, отступы слева и справа сайта являются фиксированными в 100 пикселей. Для заголовков верхний отступ является 50 пикселей. Размеры блоков новостей делятся на 2 типа, маленькие, которых большинство и большие. Размер маленьких новостных окон 383x349, размер больших новостей 785x383.

# Тестирование

Тест №1

Название проекта: Alpha-News

Рабочая версия: 0.1

Тестировщик: Костив Максим

Дата теста: 24.04.24

ID теста: TC\_UI\_1

Приоритет: Высокий

Название теста: Проверка работоспособности функционала регистрации.

Краткое изложение теста: Проверка работоспособности функционала регистрациии.

Этапы теста:

1. На главной странице сайта, нажать на кнопку “Личный кабинет”.
2. В открывшемся окне ввести данные во все имеющиеся поля.
3. Нажать на кнопку “Зарегистрироваться”.

Тестовые данные:

Логин – maxim

Пароль – maxim321

Повторите пароль - maxim321

Email – [kostiv049@gmail.com](mailto:kostiv049@gmail.com)

Телефон - 89993214252

Ожидаемый результат: Аккаунт будет зарегистрирован успешно

Фактический результат:

Аккаунт зарегистрирован в системе

Статус: Зачет

Тест №2

Название проекта: Alpha-News

Рабочая версия: 0.1

Тестировщик: Костив Максим

Дата теста: 24.04.24

ID теста: TC\_UI\_2

Приоритет: Высокий

Название теста: Проверка работоспособности функционала авторизации.

Краткое изложение теста: Проверка функционала регистрации и оповещений.

Этапы теста:

1. На главной странице сайта, нажать на кнопку “Личный кабинет”.
2. В открывшемся окне нажать на кнопку “Уже есть аккаунт?”.
3. Нажать на кнопку “Войти” с предоставленными данными.

Тестовые данные:

Логин – maxim

Пароль – maxim321

Ожидаемый результат: При правильно введенных данных в поля авторизации будет выполнен вход в личный кабинет.

Фактический результат:

При нажатии на кнопку личный кабинет открывается аккордеон с блоком регистрации. При вводе предоставленных данных в поля авторизации выполнен вход в личный кабинет.

Статус: Зачет

Тест №3

Название проекта: Alpha-News

Рабочая версия: 0.1

Тестировщик: Костив Максим

Дата теста: 24.04.24

ID теста: TC\_UI\_3

Приоритет: Высокий

Название теста: Проверка работоспособности функционала оповещений об ошибках при регистрации.

Краткое изложение теста: Проверка функционала оповещений при регистрации по нажатию кнопки

Этапы теста:

1. На главной странице сайта нажать на кнопку “Личный кабинет”.
2. В поля регистрации введите предоставленные данные
3. Нажмите на кнопку “Зарегистрироваться”

Тестовые данные:

В поле логин:

1. Ввести пустую строку
2. F
3. Maxx

В поле пароль:

1. Ввести пустую строку
2. F
3. Maxx321

Ожидаемый результат:

Логин:

1. Будет выведено диалоговое окно “Поле логин не заполнено”
2. Диалоговое окно “Поле логин меньше 4 символов”
3. Ошибка с логином пропадет, обработчик переключится на поле пароль.

Пароль:

1. Поле пароль не заполнено
2. Поле пароль меньше 6 символов
3. Обработчик переключится с поля пароль, на поле повторите пароль

Аналогичный результат должен быть для всех присутствующих полей

Фактический результат:

Про вводе данных в поле логин:

1. Поле логин не заполнено
2. Поле логин меньше 4 символов
3. Обработчик переключается на поле пароль.

При вводе данных в поле пароль:

1. Поле пароль не заполнено
2. Поле пароль меньше 6 символов
3. Обработчик переключается на поле повторите пароль

Для всех оставшихся полей результат аналогичен.

Статус: Зачет

# Заключение

В результате выполнения данной курсовой работы были улучшены знания в сфере изучения языков программирования и проектирования информационных систем. Все функции успешно удалось реализовать, готова первая версия новостного сайта Alpha-News.

Проект выполнен без использования сторонних фреймворков и баз данных. Все выполнено на чистом Java Script.

# Список литературы

1. <https://learn.javascript.ru/> - современная документация для Java Script
2. <https://lyceum.urfu.ru/fileadmin/user_upload/docs/GOST_7.0.97-2016.pdf> - Система стандартов по информации, библиотечному и издательскому делу ОРГАНИЗАЦИОННО-РАСПОРЯДИТЕЛЬНАЯ ДОКУМЕНТАЦИЯ Требования к оформлению документов

# Приложение A.

Idef0-диаграмма

Изображение выглядит как текст, диаграмма, линия, Параллельный

Автоматически созданное описание

Декомпозиция

Изображение выглядит как текст, диаграмма, чек, Параллельный

Автоматически созданное описание

# Приложение Б

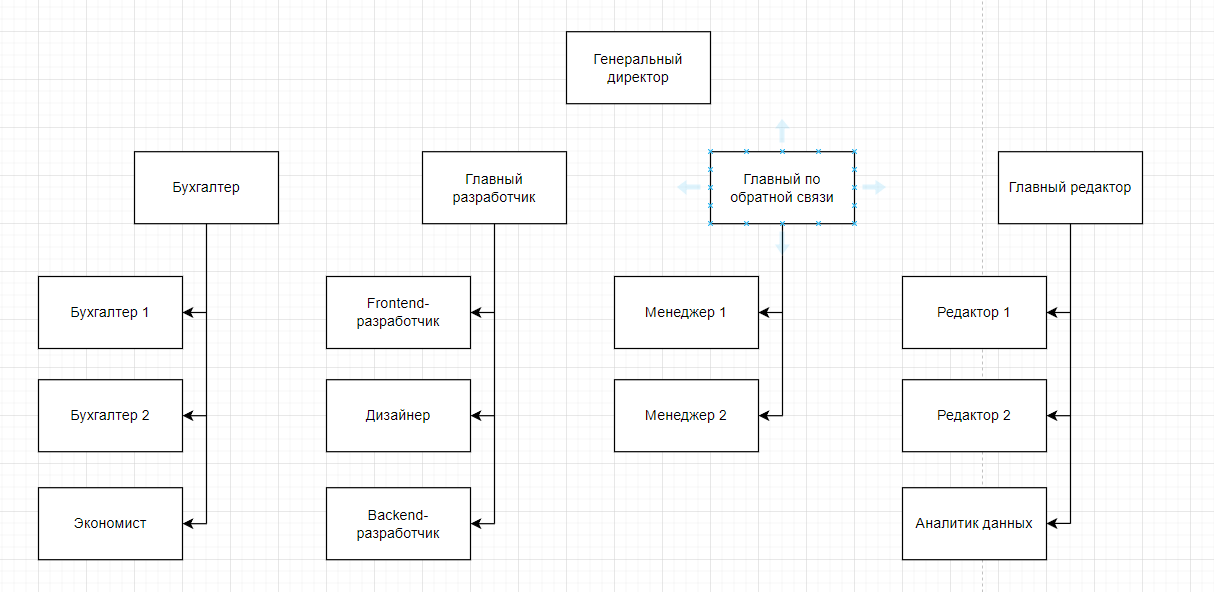
Диаграмма Ганта

Изображение выглядит как текст, линия, Шрифт, График

Автоматически созданное описание

# Приложение С

Организационная диаграмма



Приложение Д

Блок-схема окна регистрации