

УО «Белорусский государственный технологический университет»

Факультет информационных технологий
Кафедра «Программная инженерия»
Специальность 1-47 01 02 «Программная инженерия»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
КУРСОВОГО ПРОЕКТА
по дисциплине «Компьютерные языки разметки»
на тему «Веб-сайт ночной клуб Petrol »

Исполнитель
Студент 1 курса группы 10 _____ Д.Н. Воробьева
подпись, дата инициалы и фамилия

Руководитель
_____ А.В.Харланович
подпись, дата инициалы и фамилия

Курсовой проект защищен с оценкой _____

Минск 2024

СОДЕРЖАНИЕ

Введение	3
1.1 Обзор предметной области.....	4
1.2 Обзор аналогов.....	4
1.2.1 Анализ сайта GIPSY	4
1.2.2 Анализ сайта RE:PUBLIC	7
1.3 Обзор технологий и программных средств.....	9
2. Проектирование	11
2.1 Описание страниц проекта.....	11
2.2 Структурная схема проекта	13
3. Реализация проекта	17
3.1 Описание верстки.....	17
4. Тестирование	25
4.1 Кроссбраузерное тестирование	25
4.2 Валидность.....	25
Заключение.....	27
1. Список использованных источников	28
ПРИЛОЖЕНИЕ А	29
ПРИЛОЖЕНИЕ Б.....	34
ПРИЛОЖЕНИЕ В	36

Введение

В наше время ночные клубы являются популярным местом для развлечений и проведения свободного времени. Они предлагают посетителям не только возможность потанцевать под любимую музыку, но и насладиться атмосферой веселья и отдыха в ночное время. Создание веб-сайта для ночного клуба является актуальной задачей, позволяющей улучшить взаимодействие с посетителями, повысить уровень сервиса и привлечь новых клиентов.

Веб-сайт даёт возможность пользоваться инструментами интернет-маркетинга: реклама, email-рассылки и т. п. В Интернете удобнее и эффективнее проводить рекламные акции, скидки. Проще анализировать результаты и следить за статистикой посещения веб-сайта.

Цель курсового проекта: разработать полноценный веб-сайт для ночного клуба "Petrol", который будет представлять собой эффективный инструмент для взаимодействия с посетителями и продвижения клуба в онлайн-среде, с использованием HTML5, а также с применением SCSS/CSS3.

Целевая аудитория веб-сайта ночного клуба "Petrol" включает в себя широкий круг потенциальных посетителей, которые заинтересованы в разнообразных аспектах клубной жизни, начиная от музыкальных событий и заканчивая организацией частных мероприятий.

Задачи курсового проекта:

- проанализировать сайты схожей тематики;
- проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
- разработать структуру веб-сайта;
- протестировать веб-сайт.

На заключительном этапе следует проверить итоговую версию сайта на валидность и отображение в различных браузерах. После полного прохождения тестирования сайта задачи курсового проекта будут выполнены.

1. Аналитический обзор

1.1 Обзор предметной области

В наше время, когда развлекательная индустрия является важной составляющей социокультурной жизни, ночные клубы сталкиваются с рядом вызовов, связанных с обеспечением конкурентоспособности, привлечением и удержанием посетителей. Поддержание репутации и привлечение новых клиентов требует не только качественного предоставления развлекательных услуг, но и эффективной коммуникации с целевой аудиторией.

В современном мире создание и поддержание веб-сайта для ночного клуба становится необходимостью. Веб-сайт предоставляет возможность представить клуб в интернете, привлечь внимание потенциальных посетителей и обеспечить информацией о предстоящих мероприятиях, специальных акциях и услугах. Он также играет ключевую роль в установлении контакта с клиентами и поддержании взаимодействия с ними.

Анализ существующих веб-сайтов ночных клубов позволяет выявить тенденции в их оформлении и функционале. Эффективный веб-сайт для клуба должен отражать его уникальный стиль и атмосферу, предоставлять информацию о мероприятиях и ди-джеях, а также обеспечивать возможность онлайн-бронирования столов и покупки билетов.

Онлайн-сервисы и сайты играют важную роль в упрощении процесса привлечения и обслуживания клиентов. Они позволяют сократить временные затраты посетителей и обеспечивают удобство взаимодействия с клубом, что способствует увеличению его конкурентоспособности и успешности на рынке развлечений.

1.2 Обзор аналогов

Чтобы сверстать веб-сайт, следует проанализировать сайты со схожей тематикой. Аналогами послужили следующие веб-сайты:

- GIPSY[1];
- republic[2];

Тематика сайтов-аналогов – ночной клуб.

1.2.1 Анализ сайта GIPSY

Сайт ночного клуба GIPSY представляет собой стильный и современный веб-ресурс, который призван привлечь внимание посетителей и предложить им информацию о мероприятиях, атмосфере клуба и услугах.

Сайт имеет современный и привлекательный дизайн, который отражает атмосферу клуба и привлекает внимание посетителей. Использование крупных изображений, ярких цветов и стильных шрифтов на главной странице (рисунок 1.1.1) создает впечатление о модном и качественном месте для проведения времени.

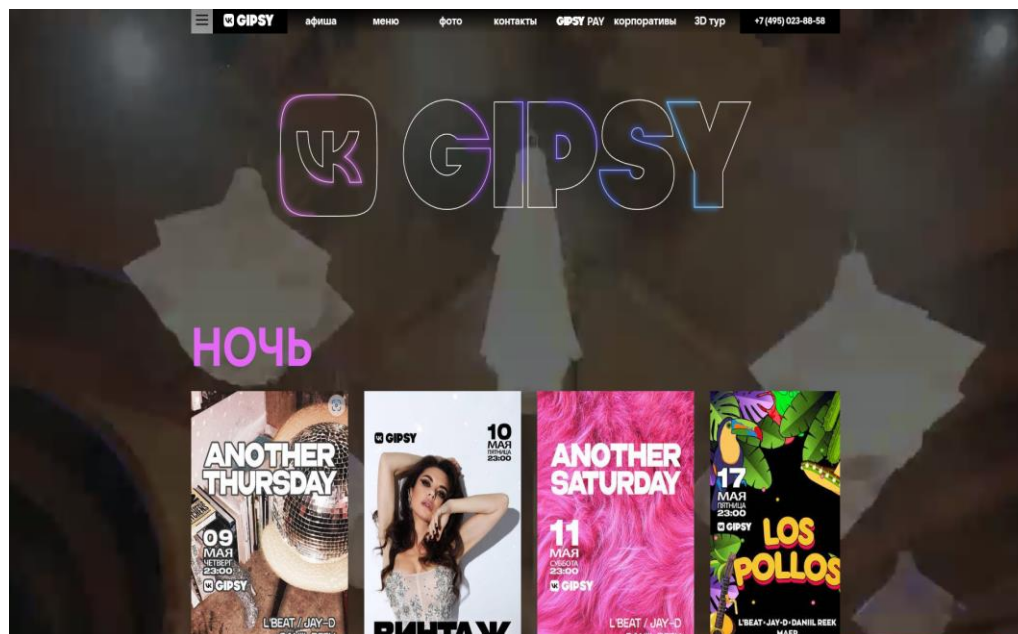


Рисунок 1.1.1 – Главная страница

Навигация по сайту интуитивно понятна и удобна для пользователей. Выпадающее меню расположено в верхнем левом углу и содержит основные разделы, такие как "Афиша", "Меню", "Фото и видео", "Контакты" и т.д. Пользователи легко могут найти нужную информацию и ознакомиться с ней.

Сайт содержит подробную информацию о предстоящих мероприятиях, включая даты, время, описание и афиши. Это позволяет посетителям быть в курсе всех актуальных событий и планировать свои вечера заранее.

Наличие фотогалереи (рисунок 1.1.2) и видеороликов позволяет пользователям получить представление о атмосфере клуба и увидеть артистов, выступающих на сцене. Это создает дополнительный интерес и мотивирует людей посетить клуб.

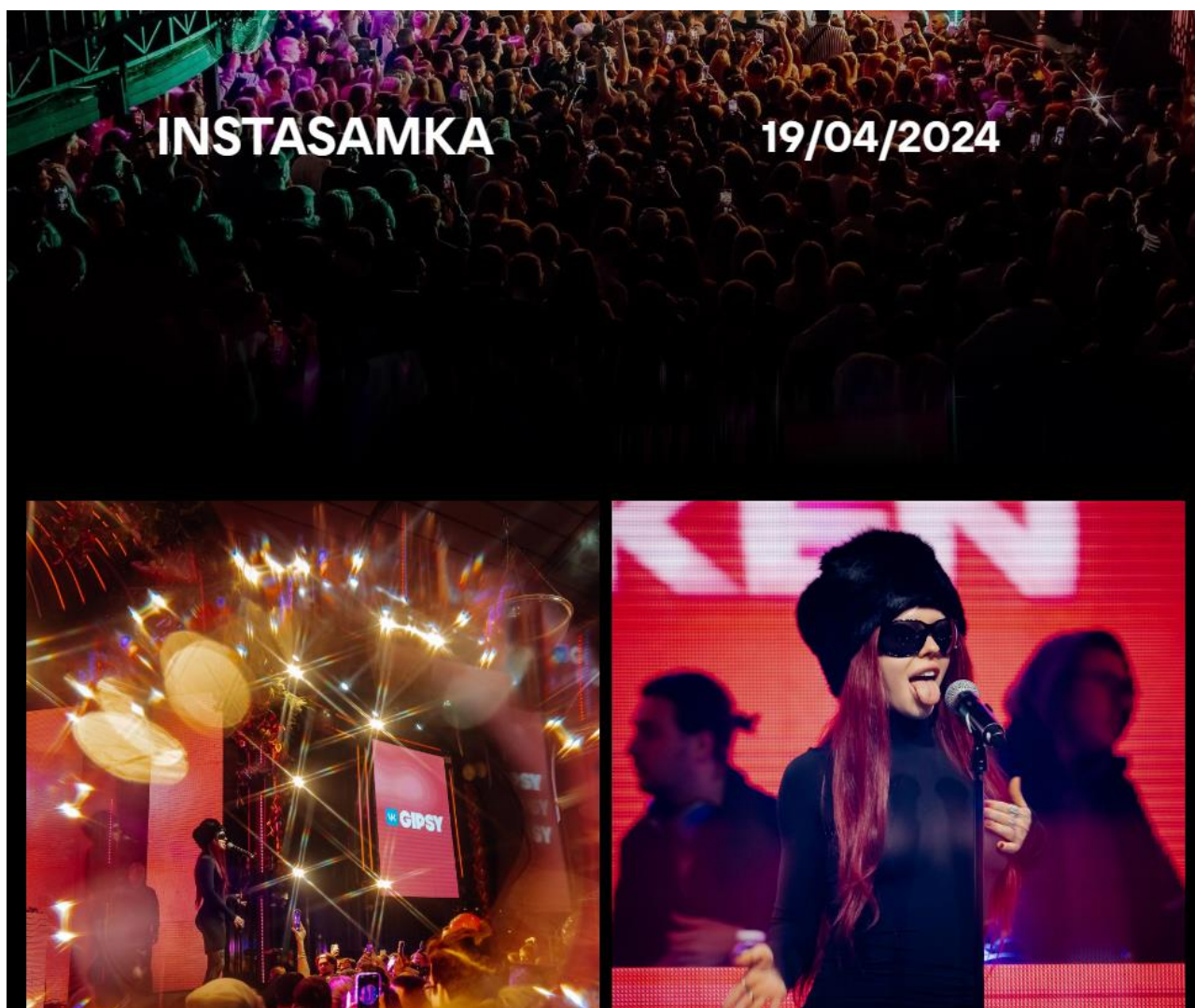


Рисунок 1.1..2 – Фотогалерея

Сайт содержит контактную информацию о клубе, включая адрес, телефон и электронную почту для связи: имеет адаптивный дизайн, который позволяет корректно отображаться на различных устройствах и экранах, включая компьютеры, планшеты и мобильные телефоны. Это обеспечивает удобство пользования сайтом независимо от устройства, которое использует посетитель. Для создания гибкой и масштабируемой верстки используются технологии Flexbox и Grid. Они позволяют эффективно распределять элементы на странице, создавать адаптивные сетки и обеспечивать удобное расположение контента. CSS3 анимации и эффекты: Для добавления интерактивности и привлекательности сайту применяются CSS3 анимации и эффекты. Это включает в себя различные переходы, изменения цветов, движение элементов и другие эффекты, которые делают сайт более привлекательным для пользователей.

HTML-код сайта организован с учетом семантических элементов HTML5. Верстка сайта выполнена с учетом кроссбраузерной совместимости, что означает его корректное отображение в различных веб-браузерах, таких как Google Chrome, Mozilla Firefox, Safari, Microsoft Edge и другие популярные браузеры.

Прямого взаимодействия с пользователями не наблюдается, нет окна регистрации, отсутствует вкладка отзывов.

1.2.2 Анализ сайта RE:PUBLIC

Сайт ночного клуба RE:PUBLIC не особо выделяется, так как у него скучное оформление и пользователю предоставляется мало взаимодействий с сайтом

Один из основных минусов сайта Republic - это его скучный дизайн. (рисунок 1.2) Визуальное оформление сайта выглядит устаревшим и малооригинальным, что может отпугнуть пользователей и снизить интерес к контенту. Отсутствие ярких цветов, оригинальных графических элементов и креативного дизайна делает сайт менее привлекательным и не запоминающимся.

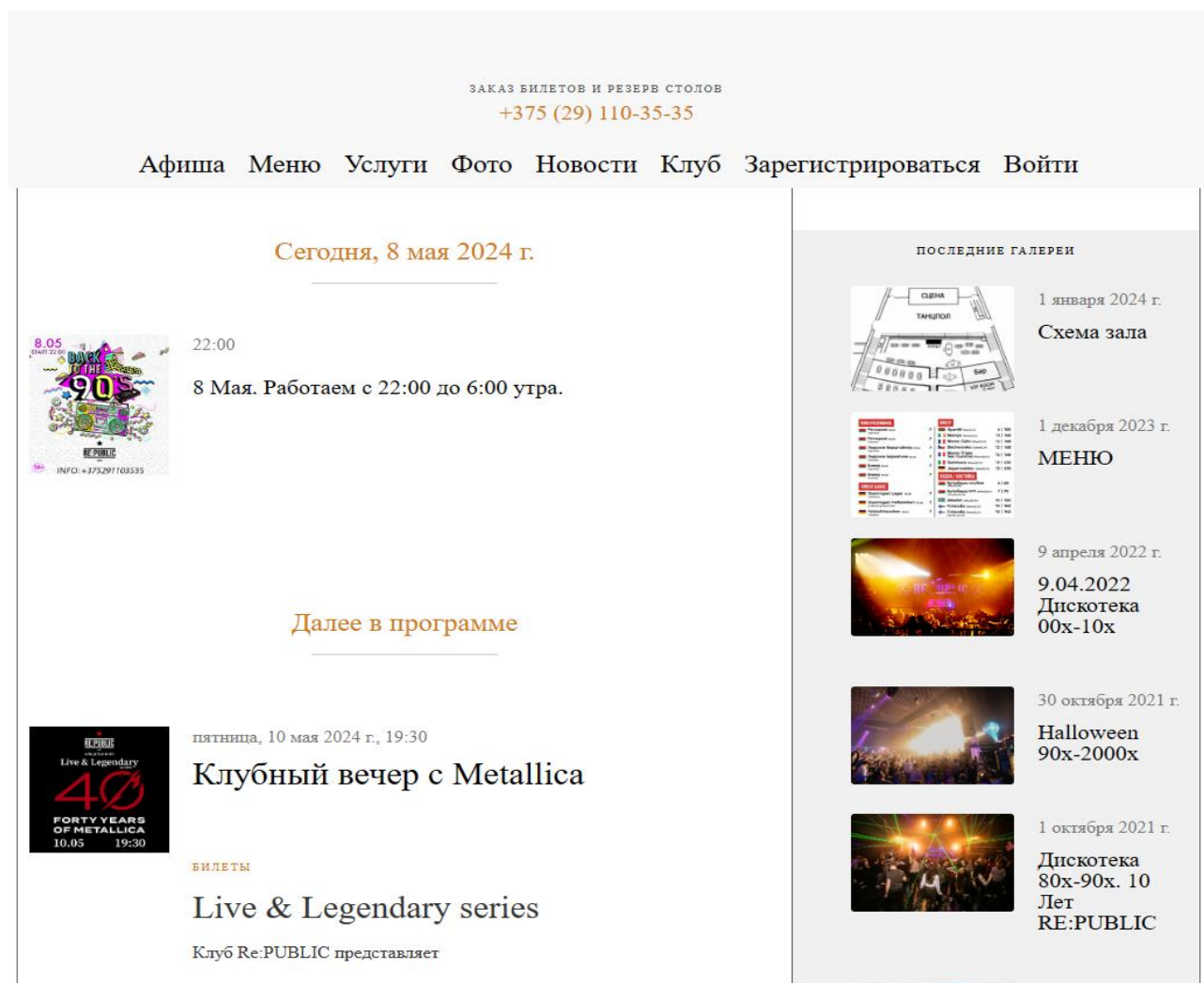


Рисунок 1.2 – Дизайн клуба

Навигация на сайте Republic чрезмерно проста и малоинформативна. Меню сайта представлено стандартными разделами, которые не всегда отражают всю доступную информацию, а категории контента могут быть недостаточно детализированными. Это затрудняет пользователям поиск нужной информации и может привести к потере интереса к сайту. Также отсутствует интерактивность и динамичность в навигации, что делает пользовательский опыт малоинтересным и малоудобным. Однако сайт содержит подробную информацию о предстоящих мероприятиях, включая даты, время, описание и афиши.

На сайте присутствует качественная фотогалерея со всех мероприятий, что говорит о ответственном подходе организаторов к рекламе и маркетингу. Сайт содержит контактную информацию о клубе, включая адрес и телефон для связи.

Верстка сайта Republic выполнена адаптивно, что обеспечивает корректное отображение контента на различных устройствах и экранах. Это позволяет пользователям комфортно использовать сайт как на десктопных компьютерах, так и на мобильных устройствах, также верстка сделана с соблюдением семантических правил, что улучшает индексацию и оптимизацию сайта для поисковых систем. Это способствует повышению видимости сайта в поисковой выдаче и привлечению новых пользователей. Страницы сайта загружаются быстро благодаря оптимизированной верстке и использованию современных технологий. Это улучшает пользовательский опыт и снижает отказы, так как пользователи не тратят много времени на ожидание загрузки страниц.

Верстка сайта Republic может отображаться неодинаково на различных браузерах из-за недостаточной кроссбраузерной совместимости. Это может привести к непредсказуемому пользовательскому опыту и ухудшить впечатление о сайте. Она не обладает достаточной интерактивностью и анимацией, что может уменьшить привлекательность сайта для пользователей. Отсутствие динамичных элементов может сделать пользовательский опыт менее увлекательным и запоминающимся.

Прямого взаимодействия с пользователями не наблюдается, нет окна регистрации, отсутствует вкладка отзывов.

1.3 Обзор технологий и программных средств

HTML (HyperText Markup Language) – стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS – это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

JavaScript это язык, который позволяет применять сложные вещи на web странице – каждый раз, когда на ней происходит что-то большее, чем просто её статичное отображение – без JavaScript не обошлось. Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять

скрипты на странице. JavaScript можно использовать не только в браузере, это полноценный язык, программы на котором можно запускать и на сервере.

Сайт будет реализован по макету, который хранится в программе Figma.

Вёрстка сайта реализуется в редакторе кода Visual Studio Code. VSCode – это бесплатное и удобное программное обеспечение, предназначенное специально для написания кода, и в частности вёрстки сайта. Для этого в программе предусмотрено множество встроенных функций, ускоряющих процесс написания кода, благодаря всплывающим подсказкам, навигации по написанному коду, автоматическому формированию отступов вложенных строк и встроенной функции автозаполнения. Visual Studio Code имеет отзывчивый и приятный интерфейс и возможность подключения огромного множества различных расширений (плагинов) отлично дополняющих основной функционал.

Будут использованы плагины «Emmet», «Live Server», «Live Sass Compiler».

Первый плагин упрощает процесс вёрстки. Для написания сложного фрагмента кода достаточно написать простое сокращение, нажать клавишу TAB, далее плагин сам скомпилирует его в нужные строки кода. Здесь используются и операторы вложенности, которые ускоряют разметку древовидных элементов.

Live Server создаёт локальный сервер и автоматически транслирует на html-страницу изменения, внесённые в код.

Веб-сайт будет реализовываться не на привычном CSS, а на SCSS. Он является «диалектом» языка SASS. Это упростит написание стилей.

Live Sass Compiler – расширение, которое в интерактивном режиме компилирует scss-файлы в css-файлы. Для этого нужно нажать клавишу Watch Sass.

2. Проектирование

2.1 Описание страниц проекта

Сайт содержит 7 страниц, где 6 – основные и последняя является страницей ошибки 404. Макеты страниц представлены в приложении на рисунке Б.1. На страницах есть повторяющиеся обязательные элементы любого сайта – хедер и футер. Они содержат прямые ссылки на остальные страницы, что обеспечивает свободный переход на любую страницу из любой точки сайта. Хедер имеет стандартную структуру. В данном проекте она содержит 3 блока. Первый – логотип и лозунг клуба, справа сверху расположена кнопка для перехода к странице с правилами клуба». Эта кнопка имеет фиксированное положение, поэтому она остается при скролле сайта. Футер на всех страницах является общим элементом. В нём содержится контактная информация клуба. Блоки расположены вертикально друг за другом, при мобильной версии блоки лишь становятся меньше. На главной странице больше всего информативных блоков. На ней находятся навигационные кнопки для перехода к следующим страницам. При открытии страницы пользователю сразу предлагают зарегистрироваться. Во всплывающем окне регистрации требуется указать имя, номер телефона, почту и пароль. После заполнения формы выскакивает короткое сообщение о том, что пользователь успешно зарегистрирован. В самом начале посещения страницы пользователь видит логотип клуба с всплывающим лозунгом, ниже – кнопки навигации, и в самом конце адрес клуба и контактную информацию. Заглавный блок и тело страницы разделены полупрозрачной линией. На странице находятся 4 главные кнопки для перехода на след страницу и самая нижняя кнопка для перехода к окну регистрации. Через кнопку Афиша мы попадаем на страницу с таким же хедером, но вместо лозунга написано название страницы. Ниже расположен календарь всех мероприятий за ближайший месяц. Календарь реализован через JavaScript. На остальных страницах похожая реализация блоков: Вместо кнопки с правилами клуба появляется кнопка для возвращения на главную страницу, вместо лозунга – название страницы, на которой вы находитесь. На странице “меню и цены” меню продукции реализовано как пять выпадающих бургер-меню. При нажатии на кнопку “заказать столик” появляется схема клуба, а ниже форма для заполнения и заказа посадочного места. Последняя из навигационных кнопок – переход на страницу с отзывами. На странице можно просмотреть комментарии, оставленные посетителем. Имеется рейтинговая система со звездами, подсчет звезд с комментарием осуществляется с помощью JavaScript. Последняя страница об ошибке 404 – страница, в которой показывает сообщение об

ошибке. На ней находится хэдер как на главной. Самой маленькой является страница «404». На неё осуществляется переход с ссылок, для которых нет отдельных страниц. Содержит шапку и изображение. Изображение страницы представлено на рисунке 2.1. Под текстом расположена кнопка «Вернуться на главную», это задержит пользователя на странице.

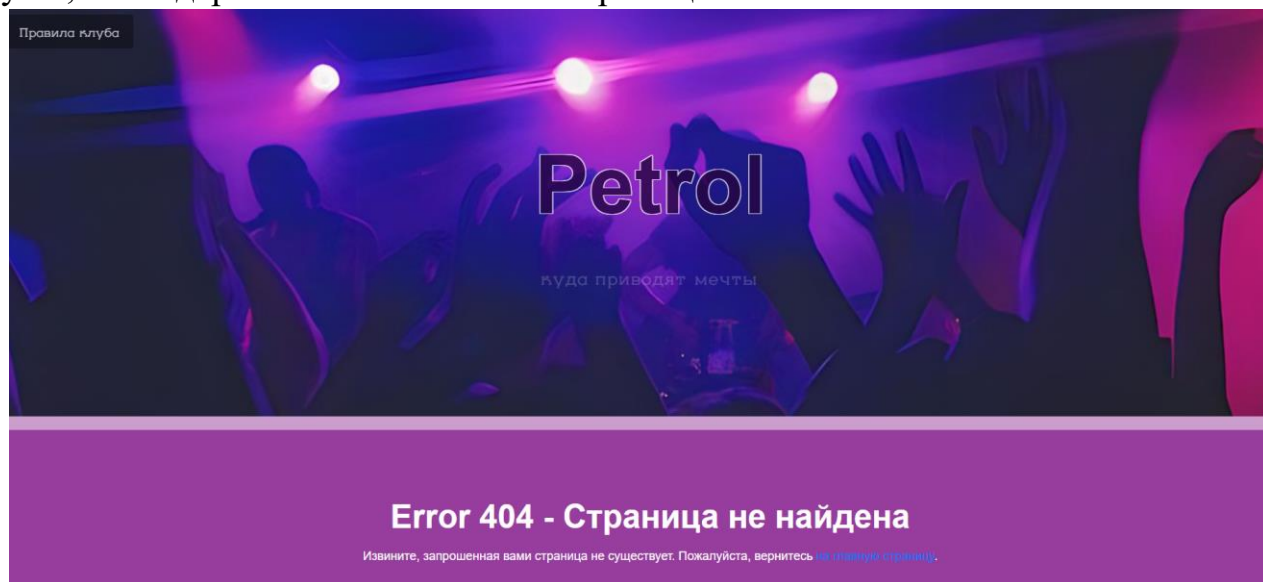


Рисунок 2.1 – Страница с ошибкой

Основные цвета сайта – чёрный, фиолетовый и белый. В большинстве блоков преобладает тёмный фон. Он имеет ряд преимуществ:

- поддержка визуальной иерархии;
- глубина визуального представления контента;
- ассоциации с элегантностью и таинственностью;
- престижный внешний вид;
- внешняя привлекательность.

Для контраста информационных блоков используется переход фона с тёмного на светлый. Тем самым пользователь разграничивает информацию. Для блока шапки используется темное изображение, граничащее с фиолетовым телом страницы. Это придаёт стилистическое единство частям страницы. Сочетание чёрного и фиолетового относится к цветам нейтральной гаммы. Но использование только этих 2 цветов придаёт мрачный вид. Поэтому в основной палитре сайта имеется третий цвет – белый. Он является контрастным к чёрному, поэтому они хорошо сочетаются на странице. С тёмным фоном грамотно и уместно смотрятся шрифты без засечек. Они имеют высокую читаемость на чёрном. Поэтому в проекте для основного текста используется геометрический шрифт « Anaheim ». Пример текстового блока приведён в приложении на рисунке Б.2 Для заголовков используется противоположный по

характеристикам шрифт с засечками – « KyivTypeSerif-VarGX ». Пример в приложении на рисунке Б.3. В блоках с тёмным или фиолетовым фоном используется белый цвет шрифта. Для светлого фона – тёмно-фиолетовый. Не используется чисто чёрный, так как данное сочетание больше напрягает зрение, чем выбранное. При наведении на все кнопки происходит анимация. Кнопка увеличивается, размер шрифта текста тоже и текст становится непрозрачным. Кнопка для правил не увеличивается, а лишь перестает быть прозрачной

Пример анимации кнопок представлен на рисунке 2.2.

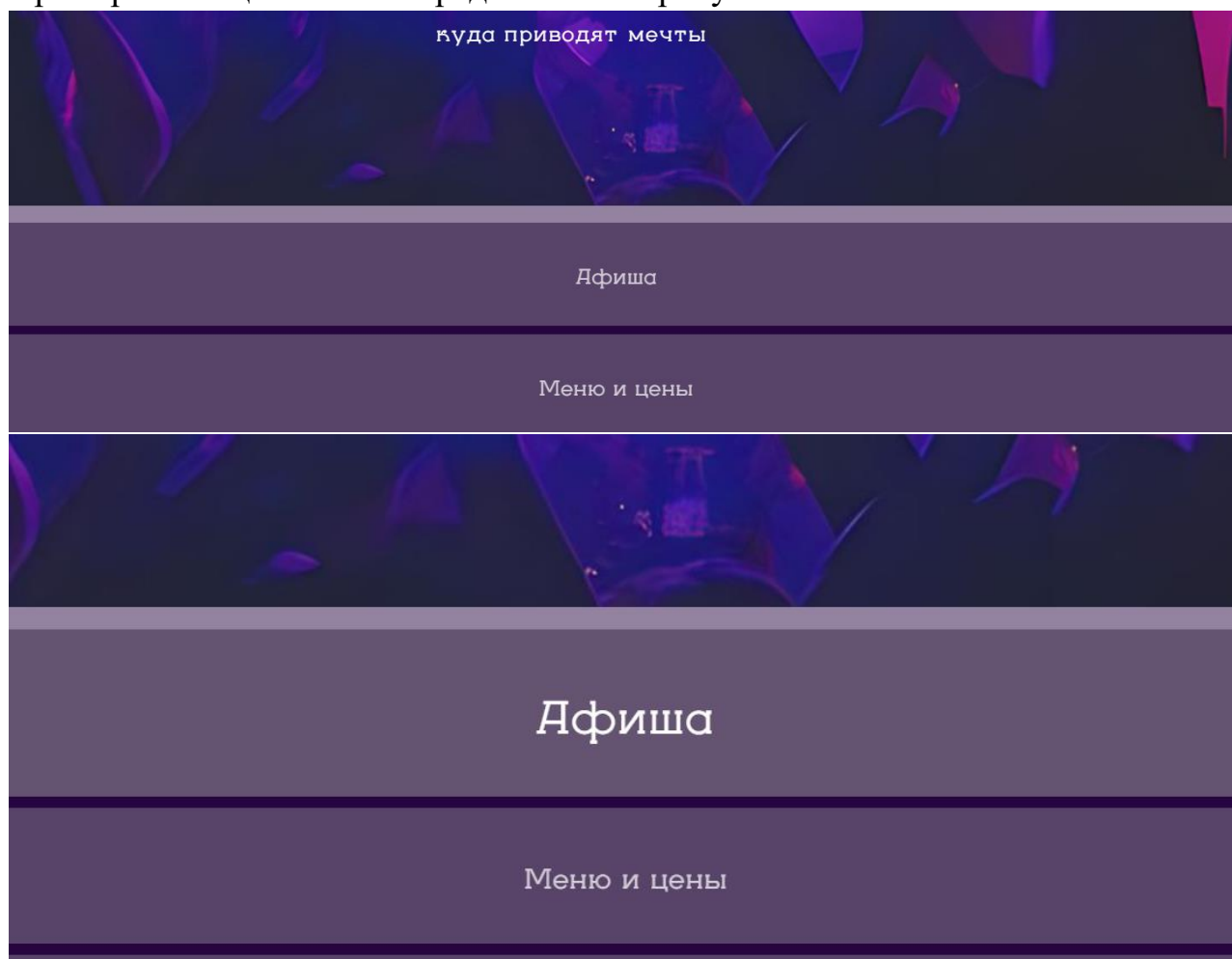


Рисунок 2.2 – Анимация кнопок при наведении

2.2 Структурная схема проекта

Структурная схема проекта определяет основные функциональные составляющие сайта и помогает продемонстрировать взаимосвязь элементов. Сайт является многостраничным. Такой подход к структуре сайта является обоснованным. Многостраничные сайты имеют ряд преимуществ:

- масштабируемость контента;

- отличный SEO-потенциал;
- наглядность для пользователя.

Структура сайта – простая, трёхуровневая. Страницы можно разделить на 3 условных уровня: главная, все услуги, конкретная услуга. Страницы связаны ссылками. Страницу ошибки можно вынести отдельно, так как она не содержит информацию о компании. На сайте имеется несколько страниц: «Главная», «Афиша», «Меню и цены», «Заказ столов», «Правила клуба», «Отзывы и поддержка», «404». С главной страницы осуществляется переход на все остальные. Это можно осуществить при нажатии на соответствующий блок, из шапки или футера. Находясь на остальных страницах возможен переход между ними. Страница «404» открывается при нажатии на ссылку, для которой нет отдельной страницы. В Приложении на рисунке Б.4 приведена карта сайта. Для создания сайта была создана папка со всеми файлами, нужными для функционирования проекта. Проект содержит 4 основные расширения (рисунок 2.3).







































Имя	Дата изменения	Тип	Размер
 fonts	27.04.2024 22:01	Папка с файлами	
 404.html	08.05.2024 22:24	Microsoft Edge H...	2 КБ
 afisha.html	28.04.2024 21:40	Microsoft Edge H...	2 КБ
 club.html	07.05.2024 15:37	Microsoft Edge H...	3 КБ
 menu.html	07.05.2024 15:27	Microsoft Edge H...	11 КБ
 order.html	28.04.2024 21:40	Microsoft Edge H...	4 КБ
 rules.html	28.04.2024 21:40	Microsoft Edge H...	4 КБ
 sup.html	28.04.2024 22:16	Microsoft Edge H...	6 КБ
 1.pdf	18.03.2024 02:17	Microsoft Edge P...	1 171 КБ
 Untitled.pdf	18.03.2024 02:17	Microsoft Edge P...	1 171 КБ
 anaheim.zip	27.04.2024 21:52	Архив ZIP - WinR...	28 КБ
 kyiv-type-serif.zip	27.04.2024 21:40	Архив ZIP - WinR...	890 КБ
 # afisha.css	28.04.2024 21:40	Исходный файл ...	5 КБ
 # club.css	28.04.2024 21:40	Исходный файл ...	11 КБ
 # menu.css	28.04.2024 21:40	Исходный файл ...	12 КБ
 # order.css	28.04.2024 21:40	Исходный файл ...	6 КБ
 # rules.css	28.04.2024 21:40	Исходный файл ...	6 КБ
 # sup.css	28.04.2024 22:23	Исходный файл ...	7 КБ
 JS cal.js	28.04.2024 21:40	Исходный файл J...	4 КБ
 JS code.js	28.04.2024 22:19	Исходный файл J...	3 КБ
 JS menu.js	28.04.2024 20:32	Исходный файл J...	0 КБ
 JS order.js	28.04.2024 21:40	Исходный файл J...	4 КБ
 order.txt	28.04.2024 17:05	Текстовый докум...	0 КБ
 6f20e416e09403297f81efa3b9c40dca.jpg	19.03.2024 00:15	Файл "JPG"	280 КБ
 1812fdf1855b185ed035dd3589afc277.jpg	19.03.2024 00:15	Файл "JPG"	144 КБ
 91797ca0e23d52c3a4cf6582aa0b2428.jpg	19.03.2024 00:15	Файл "JPG"	162 КБ
 1710715744024.jpg	18.03.2024 01:49	Файл "JPG"	56 КБ
 b5800bd64bbea7d4c47ddae4beb37e50.jpg	19.03.2024 00:21	Файл "JPG"	122 КБ
 bd48a85be482e3868beb6449f2dad682.jpg	19.03.2024 00:21	Файл "JPG"	192 КБ
 photo_1_2024-03-18_01-44-56.jpg	18.03.2024 01:44	Файл "JPG"	26 КБ
 photo_2_2024-03-18_01-44-56.jpg	18.03.2024 01:44	Файл "JPG"	25 КБ
 photo_3_2024-03-18_01-44-56.jpg	18.03.2024 01:44	Файл "JPG"	38 КБ
 photo_4_2024-03-18_01-44-56.jpg	18.03.2024 01:44	Файл "JPG"	16 КБ
 photo_5_2024-03-18_01-44-56.jpg	18.03.2024 01:44	Файл "JPG"	66 КБ
 photo_2024-03-18_02-29-23.jpg	18.03.2024 02:29	Файл "JPG"	92 КБ
 1стр.png	18.03.2024 02:19	Файл "PNG"	194 КБ
 2стр.png	18.03.2024 02:43	Файл "PNG"	308 КБ
 курсовой.rtf	08.05.2024 20:23	Формат RTF	26 995 КБ

Рисунок 2.3 – Файлы проекта

Каждый из них включает соответствующие названию файлы, нужные для полноценного функционирования проекта. Папка содержит 7-html файлов. Главная страница и ещё 6, которые называются, исходя из их содержимого – «Главная», «Афиша», «Меню и цены», «Заказ столов», «Правила клуба», «Отзывы и поддержка», «404». Также папка содержит 6 css файлов. В папке fonts содержатся все нужные файлы для настройки шрифтов. В папке находятся img файлы, содержащие все изображения проекта. Для их вставки в код используются 2 способа: задним фоном – background и через тег . Файлы js содержат перечень всех используемых скриптов.

3. Реализация проекта

3.1 Описание верстки

Сайт реализован с помощью препроцессора SCSS, который компилирует в CSS. Данная технология даёт дополнительные возможности для написания кода. Здесь можно использовать переменные, вложенные селекторы, медиазапросы, собирать несколько файлов в один, миксины, наследование. SCSS пишется в отдельных файлах с расширением .scss. После этого нужно подключить css к html. На рисунке 3.1 изображено подключение к html файлу.

```
<title>Клуб "Petrol"</title>
<link rel="stylesheet" type="text/css" href="club.css">
</head>
```

Рисунок 3.1 – Подключение CSS В теге

В теге <body> содержится контент страницы, отображаемый в окне браузера. Каждый смысловой блок обернут в тег или в семантические теги, такие как <header> и <footer>. Последнее делается, чтобы в дальнейшем избежать путаницы в ориентировании в разметке и использовать в остальных страницах. Каждый крупный блок содержит обёртку <div class="container">. Для данного класса прописаны стили, которые устанавливают размеры содержимого, чтобы блок не выбивался из потока. Центрируется блок с помощью свойства margin: 0 auto. Ширина контейнера устанавливалась исходя из макета. Для конкретного расширения экрана значения меняются в меньшую сторону через медиазапросы. Устанавливаются внутренние отступы, чтобы избежать прилипания контента к краям. Снимок экрана из редактора кода приведён на рисунке 3.2.

```
.container {
  width: 90%;
  margin: 0 auto;
  padding-top: 20px;
  padding-bottom: 20px;
  color: #fff;
  text-align: center;
}
```

Рисунок 3.2 – Класс «container»

С помощью тега <header> была создана шапка сайта для всех страниц.

В начале указываются основные параметры элемента: text-align: center;, чтобы содержимое было выровнено по центру; width: 100vw;, чтобы ширина элемента была равна ширине видимой области окна браузера; height: 600px;, чтобы задать фиксированную высоту в 600 пикселей; и position: relative;, указывающее, что элемент будет позиционироваться относительно его изначального местоположения.

Последующий блок стилей применяется к псевдоэлементу ::after элемента <header> для декоративной полупрозрачной полосы. Свойство content: ' '; обязательно для псевдоэлемента, используется для абсолютного позиционирования псевдоэлемента относительно его ближайшего позиционированного родителя. bottom: -20px; устанавливает отрицательный отступ снизу на 20 пикселей, left: 0; задает позицию слева равной 0. width: 100%; определяет ширину псевдоэлемента в 100% от ширины родительского элемента, а height: 20px; устанавливает высоту в 20 пикселей. Наконец, background-color: rgba(255, 255, 255, 0.5); устанавливает цвет фона полосы, в данном случае, прозрачный белый. z-index: -1; устанавливает индекс слоя, чтобы псевдоэлемент находился под содержимым основного элемента <header>.

Фрагмент CSS-кода с псевдоэлементом представлен в листинге 3.1.

```
header::after {
    content: '';
    position: absolute;
    bottom: -20px;
    left: 0;
    width: 100%;
    height: 20px;
    background-color: rgba(255, 255, 255, 0.5);
    z-index: -1;
}
```

Листинг 3.1 – использование псевдоэлемента after

Такая полоска реализована на всех страницах.

Для создания кнопок на веб-странице используются гиперссылки <a> внутри контейнера с классом .buttons-container. Стилизация кнопок определена в CSS селекторе .buttons-container a. Каждая кнопка имеет определенные размеры,

цвет фона с прозрачностью для эффекта, а также настройки шрифта для удобства чтения. При наведении курсора на кнопку происходит плавное изменение размера шрифта и прозрачности, что делает интерактивный эффект более привлекательным. Для кнопки с классом `.open-button` стили определяются отдельно, учитывая ее уникальный вид. Общие эффекты изменения прозрачности и размера шрифта при наведении сохраняются и для этой кнопки.

```
.buttons-container {
    display: flex;
    justify-content: space-between; /* Равномерное распределение
кнопок по ширине */
    padding: 20px 0; /* Внешний отступ между декоративной полоской
и кнопками */
    text-align: center; /* Выравнивание текста по центру */
}

.buttons-container a{
    display: block; /* Превратить ссылки в блочные элементы */
    width: 100%; /* Занимать всю доступную ширину */
    padding: 40px 50px; /* Увеличиваем отступы */
    background-color: rgba(162, 162, 162, 0.5); /* Прозрачный
черный цвет */
    color: white;
    font-family: 'CustomFont', sans-serif;
    opacity: 0.8; /* Прозрачность текста */
    font-size: 36px;
    border: none;
    text-decoration: none; /* Удаление подчеркивания ссылки */
    border-radius: 5px;
    cursor: pointer;
    margin: 0 10px; /* Добавляем отступы по горизонтали между
кнопками */

    transition: font-size 0.3s ease; /* Плавное изменение размера
шрифта */
}

    .buttons-container a:hover {
        font-size: 60px; /* Увеличиваем размер шрифта при
наведении */
        opacity: 1;
    }

.open-button {
```

```

        top: 50px;
        left: 50px;
        padding: 20px 40px; /* Увеличиваем отступы */
        color: white;
        border: none;
        border-radius: 10px; /* Увеличиваем радиус скругления
углов */
        cursor: pointer;
        font-family: 'CustomFont', sans-serif;
        font-size: 26px;
        text-decoration: none; /* Удаление подчеркивания ссылки
*/
        opacity: 0.7; /* Прозрачность кнопки */
        background-color: rgba(0, 0, 0, 0.5); /* Прозрачный
черный фон кнопки */
        transition: opacity 0.3s ease; /* Плавное изменение
прозрачности при наведении */
    }

    .open-button:hover {

        opacity: 0.9; /* Увеличение прозрачности при наведении */
    }

```

Листинг 3.2 – Реализация кнопок

В ходе работы над проектом были созданы 4 JavaScript файлов практически для всех страниц сайта, где расположены функции, отвечающие за различные аспекты работы. Функции разделены по различным файлам для упрощения навигации. В дополнение ко всему, разделение является гарантией бесперебойной работы сайта, так как ошибки, произошедшие в одном файле, не повлияют на выполнение скриптов из других файлов. Интересным аспектом стала реализация окна регистрации через JS. В реализации окна регистрации задействован скрипт, который представлен в листинге 3.3.

```
document.addEventListener('DOMContentLoaded', function() {
    const overlay = document.getElementById('overlay');
    const openButton = document.getElementById('open-button');
    const closeButton = document.getElementById('close');
    const form = document.getElementById('registration-form');
    const savedFormData = localStorage.getItem('formData');
});
```

Листинг 3.3 – Реализация окна регистрации на языке JavaScript

При загрузке DOM дерева скрипт находит элементы, управляющие оверлеем (#overlay), кнопкой открытия окна (#open-button) и кнопкой закрытия (#close). Функции openOverlay() и closeOverlay() отвечают за отображение и скрытие окна регистрации соответственно.

Для сохранения данных формы используется функция saveFormData(), которая собирает данные из полей формы (имя, email, пароль, телефон) и сохраняет их в локальном хранилище браузера в формате JSON. Обработчики событий добавлены для кнопок открытия, закрытия окна и отправки формы. Они вызывают соответствующие функции для обработки этих событий.

При загрузке страницы скрипт проверяет наличие сохраненных данных в локальном хранилище и, если они найдены, автоматически заполняет поля формы. Также данные сохраняются перед переходом на другую страницу, чтобы пользовательский ввод не был потерян.

Во втором файле JS – код реализации календаря событий. Скрипт начинается с прослушивания события DOMContentLoaded, чтобы гарантировать, что скрипт выполнится после загрузки DOM дерева.

Главная функция generateCalendar() создает календарь на основе текущего месяца. Для этого она определяет текущую дату и месяц, а также количество дней в текущем месяце. Затем она создает строки и ячейки для календаря, заполняя их числами дней месяца и информацией о событиях или случайными именами диджеев.

Информация о событиях хранится в объекте events, где ключами являются числа дней месяца, а значениями - текстовые описания событий.

Ячейки календаря заполняются в соответствии с данными из объекта events. Если для определенного дня есть событие, оно отображается в ячейке. В противном случае отображается случайное имя диджея.

Созданные элементы добавляются в таблицу календаря в соответствующем теге <tbody>. Реализация кода указана в листинге 3.4

```
document.addEventListener('DOMContentLoaded', function() {
    const calendarBody = document.querySelector('#calendar
tbody');

    // События для разных дней
    const events = {
        3: "SODA LUV 22:00-04:00",
        7: "Mayot 21:00-01:00",
        15: "RISK 20:00-02:00",
        20: "Fonkguy 18:00-03:00"
    };
    generateCalendar();
});
```

Листинг 3.4 – Реализация календаря на языке JavaScript

На сайте есть возможность заказать столик, для этого также создан код на языке JavaScript. Он реализовывает мини-магазин, который предлагает билеты и столики. Он обеспечивает взаимодействие пользователя с веб-страницей, позволяя добавлять выбранные товары в корзину, удалять их оттуда и осуществлять оплату.

Центральным элементом является объект `cart`, который отслеживает выбранные покупки пользователя. Он состоит из двух массивов: `tickets` для билетов и `tables` для столиков.

Функция `addToCart()` вызывается при добавлении товара в корзину. Она получает тип товара (билет или столик) и выбранную опцию из выпадающего списка, а затем добавляет соответствующий объект в массив `cart`.

Функция `removeFromCart()` удаляет товар из корзины по запросу пользователя. Она принимает тип товара и индекс товара в массиве корзины и удаляет соответствующий элемент.

Функция `updateCart()` обновляет отображение корзины на веб-странице. Она перерисовывает список билетов и столиков, отображая текущее содержимое корзины.

Обработчик события отправки формы оплаты (`form.addEventListener("submit")`) позволяет пользователю осуществить оплату. Он проверяет корректность заполнения всех полей формы, отправляет данные на сервер для обработки платежа, а затем выводит сообщение об успешной оплате и очищает корзину. Реализация данного кода представлена на листинге 3.5

```

let cart = {
    tickets: [],
    tables: []
};
document.addEventListener("DOMContentLoaded", function() {
    const form = document.getElementById("payment-form");

    form.addEventListener("submit", function(event) {
        event.preventDefault();

        // Получаем данные из формы
        const cardNumber =
document.getElementById("card-number").value;
        const cardHolder =
document.getElementById("card-holder").value;
        const expiryDate =
document.getElementById("expiry-date").value;
        const cvv = document.getElementById("cvv").value;

```

Листинг 3.5 – Заказ столов и билетов на языке JavaScript

3.2 Адаптивная верстка

Адаптивность — одно из ключевых требований, предъявляемых к современным сайтам. Ресурс должен одинаково хорошо демонстрироваться на экране компьютера и смартфона. Адаптивная верстка позволяет подстраиваться любому элементу сайта под разрешение экрана, позволяя менять размер шрифта, расположение объектов, цвет и т. д. Происходит это динамически, с использованием медиа-запросов, позволяющих автоматически определять разрешение монитора, тип устройства и подставлять указанные значения. Медиа-запрос начинается с ключевого слова `@media` после которого указывается одно или несколько условий. В качестве условия указывалась 23 максимальная допустимая ширина. После составления медиа-запроса, стили, указанные в нём, будут применяться только при истине итогового условия. Был прописан медиа-запрос для экранов с `max-width: 768 px`. Пример использования медиа-запроса представлен в листинге 3.6. Здесь изменяются максимальные размеры элемента, отступы, размер шрифта.

```

@media (max-width: 768px) {
    header {
        height: auto;
    }

```

```

        header::after {
            bottom: 0;
            height: 10px;
        }
        .header-text1 {
            top: 20%;
            font-size: 60px;
        }
        .header-text2 {
            top: 45%;
            font-size: 24px;
        }
        .club-rules-btn,
        .open-button {
            padding: 10px 20px;
            font-size: 18px;
        }
        .buttons-container {
            padding: 10px 0;
        }
        .buttons-container a {
            padding: 20px;
            font-size: 24px;
            margin: 10px 0;
        }
    }

    .buttons-container {
        flex-direction: column;
        align-items: stretch;
    }
    .buttons-container a {
        margin-bottom: 10px;
    }

```

Листинг 3.6 – Пример медиа-запроса

Для создания адаптивной вёрстки реализовалось несколько пунктов. Один из них – указывание единиц измерения не в px, а в процентах. Они вычисляются по формуле: ширина нужного элемента / ширина родительского элемента * 100 %. Но основная часть адаптивной. Это делает блоки резиновыми. В адаптивной вёрстке помогают flex-элементы. Выше было описано использование свойства flex-wrap в секции главной страницы. Меняем ширину родительского блока и автоматически осуществляется перенос. Пример адаптива представлен в приложении на рисунках Б5, Б6

4. Тестирование

В структуре сайта именования классам давались по технологии БЭМ. Для проверки корректности используется сервис Генератор HTML-дерева [7]. Были просканированы все 7 страниц. Ошибок не обнаружено. Максимального числа уровней сайт достигает на странице “меню и цены”– 9. Генератор так же подтвердил верность уровней заголовков. Результат представлен в приложении на рисунке В.1.

Итоговый проект был протестирован на производительность с помощью режима разработчика. Проверка осуществлялась для мобильных устройств и компьютеров. Показатель производительности для компьютеров 18.79, для телефонов 21.78. Оба показателя считаются хорошими для функционирования сайта. Все параметры сканирования приведены в приложении на рисунках В.2, В.3

4.1 Кроссбраузерное тестирование

Кроссбраузерность – важный критерий корректной работы веб-ресурса. Это инструмент поддержания посетительской лояльности, повышения конверсии и поднятия рейтинга в поисковых системах. После написания основной структуры страницы на HTML и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров. В ходе тестирования было выяснено, что все протестированные браузеры поддерживают все свойства сайта, в результате чего следует сделать вывод, что у сайта есть преимущество в кроссбраузерности.

Так как проверка на кроссбраузерность возможно на данном этапе разработке можно только ручную, некоторые свойства, которые некорректно отображаются в различных браузерах, могли остаться незамеченными.

4.2 Валидность

Валидность HTML-верстки — это её соответствие стандартам организации W3C. Отсутствие ошибок в вёрстке документа — один из

основных показателей её качества. При этом, соблюдение стандартов не обязательно для того, чтобы сайт верно отображался в различных браузерах. Сайт прошёл проверку на валидность в сервисе validator.w3.org. Ошибок не обнаружено. Результаты сканирования приведены в приложении на рисунке В.4.

Заключение

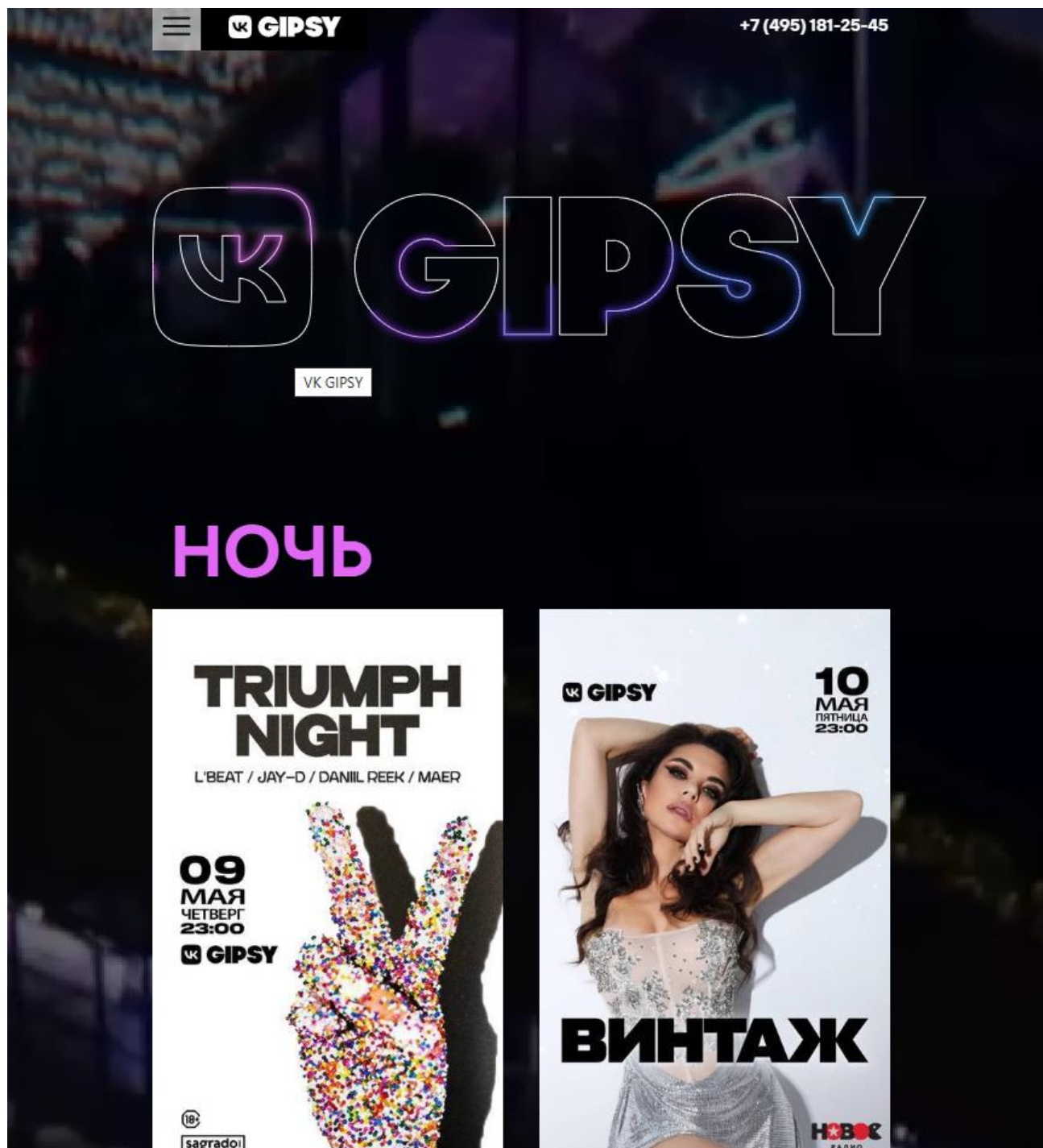
В ходе проделанной работы был свёрстан полноценный веб-сайт ночного клуба “Petrol”.

При разработке данного веб-сайта был использован широкий ряд языков разметки и таблиц стиля, который позволил увеличить функциональность и многократно повысил качество конечного продукта. Была использована адаптивная вёрстка для корректного отображения на всех устройствах. В ходе выполнения работы были рассмотрены плюсы и минусы веб-сайтов выбранной тематики. Выяснилось, каких тенденции при создании веб-сайта стоит придерживаться. Были проанализированы существующие языки разметки, инструменты и библиотеки для создания веб-сайта. В ходе данного анализа выбраны нужные методы и способы для реализации проекта. Вёрстка осуществлялась по макету из программы Figma. Стили сайта прописаны в CSS файлах. Это упрощает синтаксис кода. Для разработки некоторых деталей были использованы дополнительные технологии, не изучаемые на данном курсе, такие, как JavaScript. В результате выполнения курсового проекта был создан кроссбраузерный адаптивный сайт, состоящий из нескольких HTML-страниц. В ходе конечного тестирования была продемонстрирована кроссбраузерность и адаптивность созданного веб-сайта. Проект имеет хорошие показатели при анализе, это подтверждает выполнение поставленных задач.

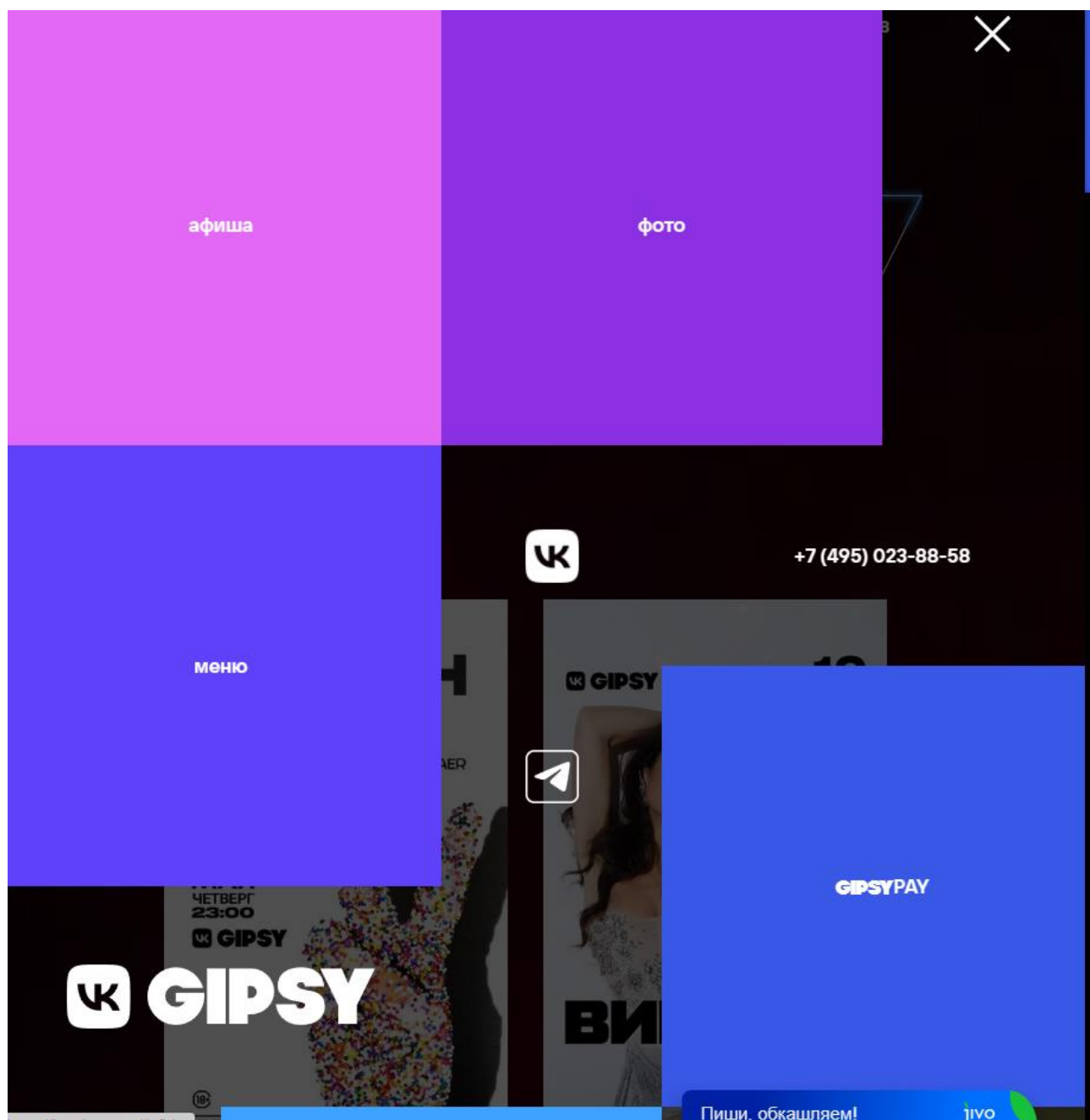
1. Список использованных источников

2. RE:PUBLIC [Электронный ресурс] – Режим доступа: <https://www.republic-club.by/> - Дата доступа: 05.03.2024
3. GIPSY [Электронный ресурс] – Режим доступа: <https://bargipsy.com/> - Дата доступа: 05.03.2024
4. ANIMA / Ночной клуб м.Менделеевская [Электронный ресурс] – Режим доступа: clubanima.ru) – Дата доступа: 05.03.2024.)
5. Генератор HTML-дерева [Электронный ресурс] – Режим доступа: <https://yoksel.github.io/html-tree/> – Дата доступа: 05.03.2024.
6. Markup Validation Service [Электронный ресурс] – Режим доступа: <https://validator.w3.org> – Дата доступа: 05.03.2024
7. DOM: работа с HTML-страницей [Электронный ресурс] – Режим доступа: <https://javascript.ru/tutorial/dom> – Дата доступа: 05.03.2024)
8. ANIMA / Ночной клуб м.Менделеевская [Электронный ресурс] – Режим доступа: clubanima.ru) – Дата доступа: 05.03.2024.)
9. Ночной клуб «Сова» [Электронный ресурс] – Режим доступа: <https://sovaclub.ru> - Дата доступа: 05.03.2024

ПРИЛОЖЕНИЕ А



А.1 – главная страница сайта GIPSY



А.2 –меню сайта GIPSY

**GIPSY****Адрес :**

Болотная наб., 3, стр. 2, Москва

Гостевой сервис :

+7 (495) 023-88-58

info@bargipsy.com

Корпоративные мероприятия :**sagrado**

+7 (495) 955-14-50

corp@sagradocorp.org

Operated by Sagrado Corp.

Бюро находок :

+7 (916) 306-92-69

Добро пожаловать в **VK Gipsy**: бар и ночной клуб в центре Москвы с умопомрачительным панорамным видом на набережную, вместительным танцполом и самой задорной развлекательной программой.



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "КЛУБ НА ОСТРОВЕ" Адрес: НАБЕРЕЖНАЯ
БОЛОТНАЯ Дом 3 Строение 2 Эт3 Пом 1 Ком 2, Москва, Москва, Москва, 119072, RU



ЗАКАЗ БИЛЕТОВ И РЕЗЕРВ СТОЛОВ

+375 (29) 110-35-35

Афиша

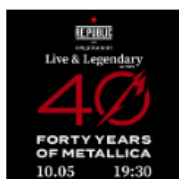
Фото

Новости

Клуб

Сегодня мероприятий нет

Завтра, 10 мая 2024 г.



19:30

Клубный вечер с Metallica



23:00

Пятница. Работаем с 23:00 до 6 утра.

Далее в программе

А.4 – главная страница сайта RE:PUBLIC

КЛУБ ОСТАВЛЯЕТ ЗА СОБОЙ ПРАВО ИЗМЕНЯТЬ И ДОПОЛНЯТЬ ТЕКУЩУЮ ПРОГРАММУ
МЕРОПРИЯТИЙ

Минск ул. Притыцкого 62. ст. м. Кунцевщина. Телефон +375 29 110 35 35.
Работаем в дни проведения концертов с 17.00 до 22.00, в дни проведения дискотек с 23.00 до 06.00.

Юридическая информация:

Индивидуальный предприниматель Старцев Андрей Валентинович
Свидетельство №190462465 выдано Минским горисполкомом 30 января 2015

Адрес: 220140 г. Минск, ул. Притыцкого, дом 62 - 109

УНП 190462465

Р/с 3013239400013 ЗАО «МТБанк», код 117

220033, г. Минск, пр. Партизанский, д. 6А

тел/факс +375 17 290 86 90

Все материалы © RE:PUBLIC. При цитировании ссылка на сайт обязательна.

© republic-club.by 2012–2024 г.



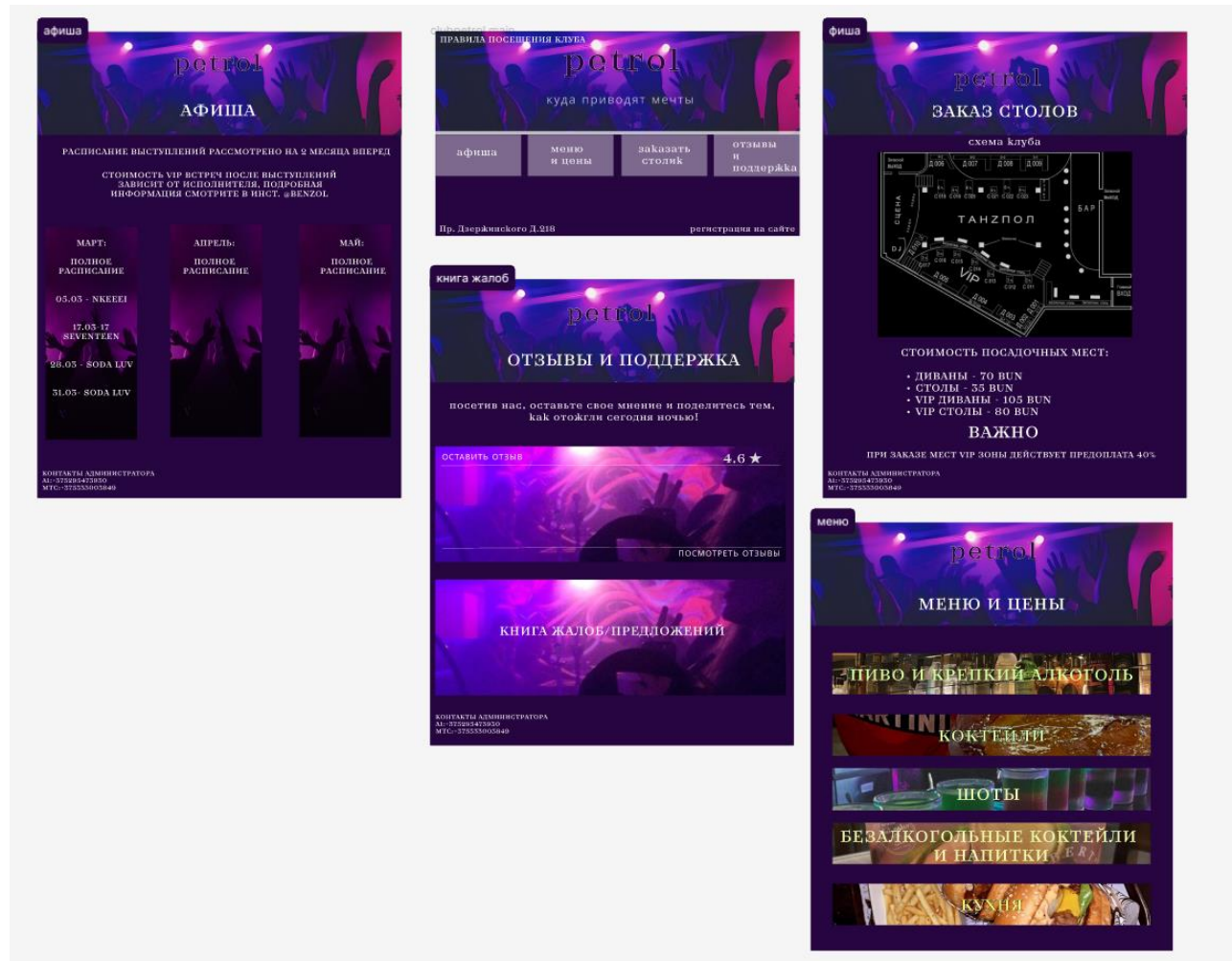
Договор публичной оферты

Правила клуба

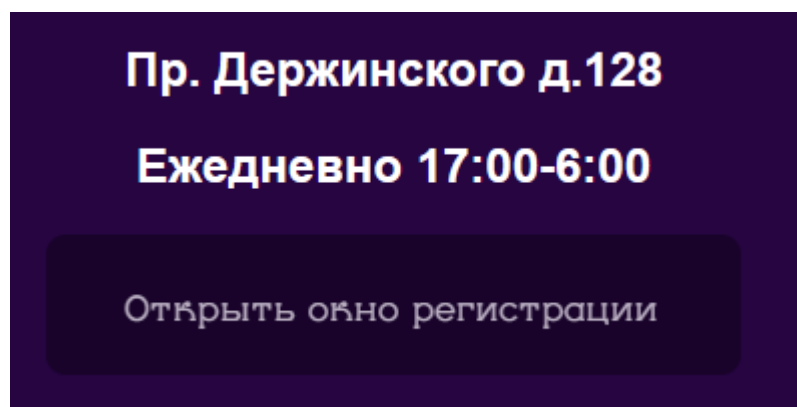
Правила оплаты

А.5— футер сайта RE:PUBLIC

ПРИЛОЖЕНИЕ Б



Б.1 – Макет сайта



Б.2 – пример текстового блока сайта



Б.3 – Шрифт заголовка

Введите код:

```
<li>Шоколадный фондан с мороженым
25 byn</li>
<li>Фруктовый салат с мятным сиропом
4 byn</li>
</ul>
</nav>
</div>
</body>
</html>
```

Уровни: 9

Структура заголовков:

H1 Petrol

- H2 Водка:
- H2 Ром:
- H2 Текила:
- H2 Виски:
- H2 Джин:
- H2 Коньяк:
- H2 Ликеры:
- H2 Абсент:
- H2 Пиво:

Структура разметки:

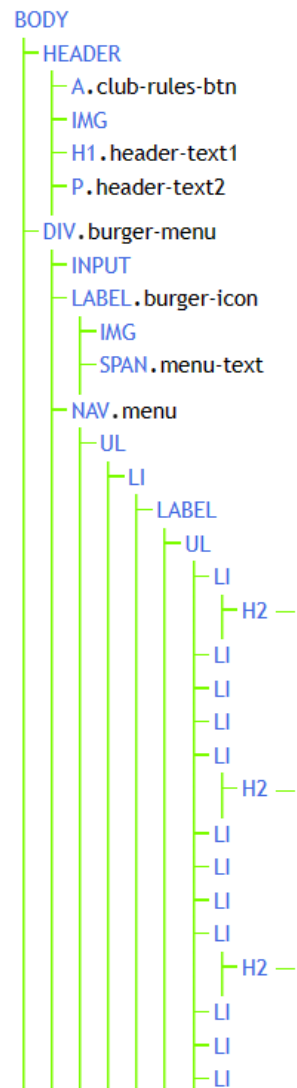


Рисунок В.1 – Генератор HTML-дерева

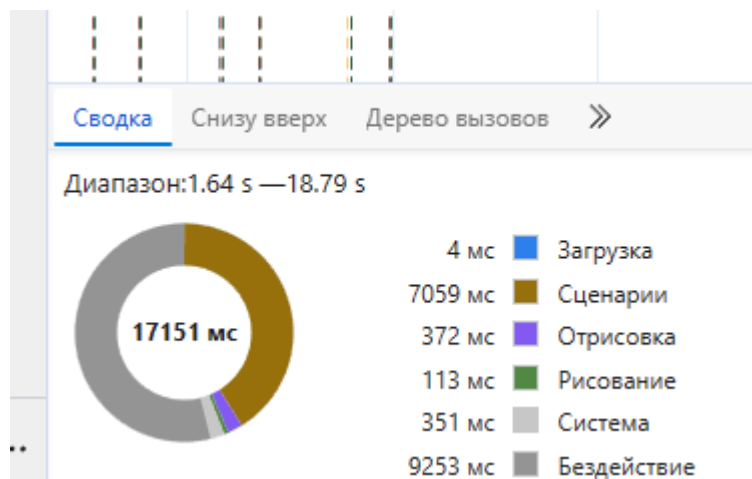


Рисунок В.2 – проверка производительности сайта

Рисунок В.3 – Проверка на валидность