

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

канд. техн. наук, доцент
должность, уч. степень, звание

подпись, дата

О.И. Красильникова
инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ

СОЗДАНИЕ WEB-САЙТА (SIFU - тематический сайт)
по дисциплине: «WEB-ТЕХНОЛОГИИ»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4028

подпись, дата

З.Х. Махмудов
инициалы, фамилия

Санкт-Петербург 2023

Оглавление

Введение.....	3
1. Дизайн сайта.....	4
1.1 Модель сайта.....	4
1.2 Макет сайта.....	5
1.3 Применение технологии каскадных таблиц стилей	7
1.4 Графическое оформление web-страницы	8
1.5 Создание навигационных панелей для сайта	9
2. Web-сценарии сайта на языке JavaScript.....	11
2.1 Назначение языка JavaScript.....	11
2.2 Разработка web-сценариев для сайта.....	12
3. Валидация кода и продвижение сайта.....	14
3.1 Валидация кода сайта	14
3.2 Способы продвижения сайтов	15
3.3 Используемые приемы продвижения сайта.....	15
4. Итоговой вариант web-сайта.....	16
Заключение	22
Список использованных источников	23
Приложение А.....	24

Введение

Целью данной курсовой работы является закрепление навыков создания веб-сайтов и использования современных технологий веб-дизайна. В рамках работы я решил разработать тематический веб-сайт, посвященный игре SIFU. SIFU — это игра-экшен с элементами боевого искусства, разработанная студией Slocap.

Современные технологии веб-дизайна предоставляют множество возможностей для создания привлекательных и функциональных веб-сайтов. Используя эти технологии, необходимо разработать сайт, который будет не только информативным и эстетически привлекательным, но и предложит пользователям удобный интерфейс и навигацию.

Введение на моем веб-сайте будет представлять краткую информацию о SIFU и его геймплее. Я намерен рассказать о основных особенностях игры, ее сюжете, персонажах и игровом процессе.

Целью главной страницы будет заинтересовать посетителей, предоставить им общее представление о SIFU и вызвать интерес к дальнейшему изучению игры на веб-сайте. Также введение будет служить ориентиром для пользователей, позволяя им легко найти необходимую информацию и продолжить взаимодействие с сайтом.

В следующих разделах работы я подробно рассмотрю различные аспекты процесса разработки веб-сайта, включая выбор технологий, структуру сайта, дизайн интерфейса и другие существенные аспекты.

1. Дизайн сайта

1.1 Модель сайта

Для моего тематического веб-сайта, я разработал следующую организационную модель:

1. Главная страница: Главная страница является стартовой точкой для посетителей сайта. На этой странице представлена общая информация о игре SIFU, ее сюжете и основных особенностях. Также здесь можно найти графический элемент, представляющий игру, и навигационное меню для перехода на другие страницы.
2. Страница "Враги": Эта страница посвящена врагам и противникам в игре SIFU. Здесь посетители могут узнать о различных типах врагов, их характеристиках, уникальных способностях и тактиках боя.
3. Страница "Новости": На странице "Новости" посетители могут получить последние обновления и информацию о игре SIFU. Здесь будут представлены новости о выпуске обновлений, событиях и акциях, связанных с игрой. Посетители могут быть в курсе всех актуальных событий и изменений в игре.
4. Страница "Закулисы": Эта страница предназначена для погружения посетителей в мир создания игры SIFU. Здесь можно узнать о процессе разработки. Такая страница позволяет углубиться в историю создания игры и узнать интересные факты о ее разработке

На рисунке 1 показана схема модели сайта



Рисунок 1 – Схема модели сайта

1.2 Макет сайта

Для создания макета веб-сайта, посвященного игре SIFU, я использовал приложение Figma. Figma предоставляет удобную платформу для создания дизайна и прототипирования веб-интерфейсов. Он позволяет создавать визуальные компоненты, определять расположение элементов на странице и создавать интерактивные прототипы.

Макет сайта в Figma был создан в соответствии с выбранным дизайном и организационной моделью, описанной ранее. Я использовал компонентный подход, разделяя сайт на отдельные элементы и блоки, чтобы обеспечить консистентность и повторное использование.

При выполнении курсовой работы использование Figma позволило мне создать эстетически привлекательный макет веб-сайта, определить

компоненты и их расположение на странице, а также создать интерактивные прототипы для демонстрации функциональности и навигации по сайту. Это позволило мне визуализировать конечный результат и обеспечить согласованность дизайна перед началом верстки и разработки сайта

Итоговая версия отличается от той, что представлена на рисунке 2, однако основные приемы и дизайнерские решения были использованы при проектировании веб-сайта

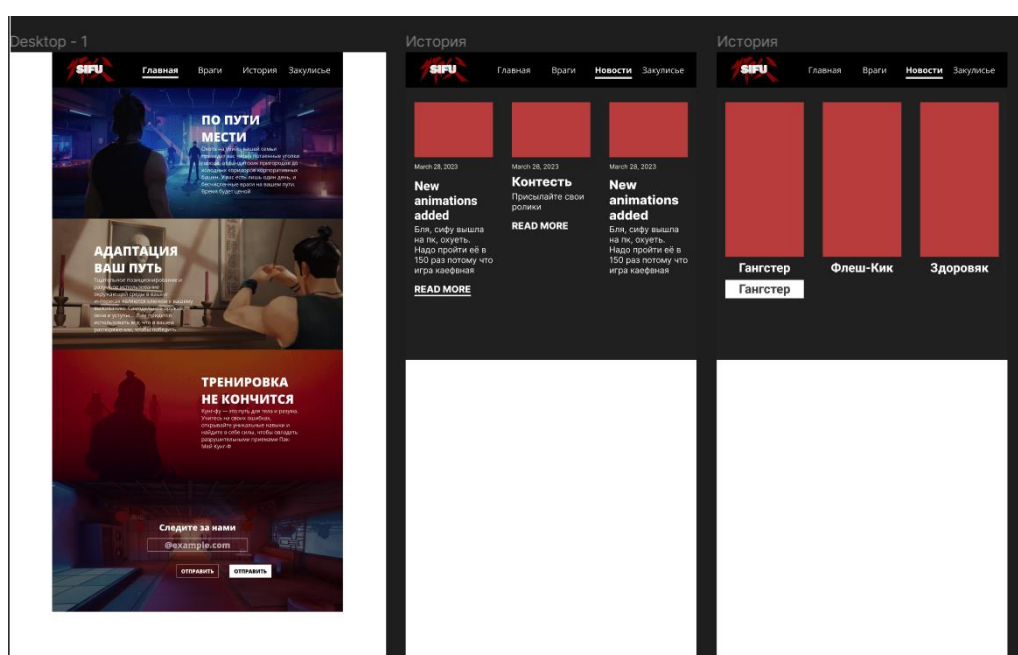


Рисунок 2 – Макет сайта

Адаптивная верстка сайта была выполнена с использованием современных технологий, таких как HTML, CSS и JavaScript. Для обеспечения отзывчивого дизайна и адаптации к различным устройствам я использовал подход Responsive Web Design (RWD), а точнее медиа-запросы

1.3 Применение технологии каскадных таблиц стилей

Технология каскадных таблиц стилей (CSS) является стандартом для описания внешнего вида веб-страниц. Суть данной технологии заключается в применении стилей к HTML-элементам, позволяя контролировать их внешний вид, расположение, размеры, цвета и другие свойства. С помощью CSS можно создавать эстетически привлекательные и современные веб-дизайны, придавая страницам уникальный стиль и макет.

В примере на рисунке 3 мы задаем стили для списка новостей. Устанавливаем значение flex, и отступы между элементами списка. Это помогает создать четкое и привлекательное разделение между новостными элементами

```
.cards-box {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: space-between;  
}  
  
.card {  
  width: 30%;  
  
  margin-left: 20px;  
  margin-right: 20px;  
  margin-bottom: 20px;  
  height: auto;  
  display: block;  
  display: flex;  
  flex-direction: column;  
  
  justify-content: flex-start;  
}
```

Рисунок 3 – Список новостей

Оптимизация кода для обеспечения кроссбраузерности включает использование вендорных префиксов для свойств, которые могут быть

неоднозначно интерпретированы разными браузерами. Пример представлен на рисунке 4

```
::-webkit-scrollbar {  
  width: 10px;  
}  
  
::-webkit-scrollbar-track {  
  background-color: transparent;  
}  
  
::-webkit-scrollbar-thumb {  
  background-color: #d6dee1;  
  border-radius: 10px;  
  border: 3px solid transparent;  
  background-clip: content-box;  
}
```

Рисунок 4 – Вендорные префиксы

1.4 Графическое оформление web-страницы

Графика играла важную роль в создании привлекательного и эстетического дизайна сайта, а также помогала визуально передавать информацию.

При выборе формата графических изображений я учитывал несколько факторов, таких как качество изображения, размер файла и поддержку форматом различных браузеров. Наиболее часто использовались форматы JPEG и PNG. Формат JPEG был выбран для фотографий и изображений с плавными переходами, а формат PNG - для графики с прозрачностью, логотипов и иллюстраций.

Оптимизация графики была одним из важных аспектов, чтобы обеспечить быструю загрузку и уменьшить потребление трафика пользователем. Некоторые приемы оптимизации графики, которые я использовал, включали сжатие изображений без потери качества с использованием инструментов сжатия, оптимизацию размеров

изображений под требуемые размеры на странице и использование форматов с наиболее эффективным сжатием для каждого конкретного типа изображения.

Также, для оформления элементов страницы или страницы в целом, я использовал приемы с фоновой графикой, чтобы придать им дополнительные эффекты и выделить их на фоне. Например, я мог использовать фоновое изображение для заголовка страницы или разделов сайта, чтобы сделать их более привлекательными и запоминающимися. Пример представлен на рисунке

Основным принципом при использовании графического оформления было сохранение баланса между эстетическими и функциональными аспектами. Графика должна была не только украшать страницу, но и поддерживать иллюстрацию и визуализацию информации, улучшать навигацию и общее восприятие контента.

1.5 Создание навигационных панелей для сайта

Требования, предъявляемые к навигации на сайте:

1. **Ясность и понятность:** Навигационная панель должна быть понятной для пользователей, чтобы они могли легко определить, какие разделы и страницы доступны для них.
2. **Компактность:** Навигационная панель должна быть компактной и занимать небольшое пространство на странице, чтобы не загромождать интерфейс.
3. **Консистентность:** Навигационная панель должна быть консистентной на всем сайте, чтобы пользователи могли легко

ориентироваться и не тратили время на изучение нового интерфейса при переходе между разделами.

Пример фрагмента кода, предназначенного для создания навигационной панели на сайте представлен на рисунке 5

```
<header>
  <div class="logo">
    
  </div>
  <nav>
    <ul>
      <li><a class="page-link" href="http://localhost:3000">Главная</a></li>
      <li><a class="page-link" href="http://localhost:3000/enemies">Враги</a></li>
      <li><a class="page-link" href="http://localhost:3000/news">Новости</a></li>
      <li><a class="page-link" href="http://localhost:3000/behind_scene">Закулисье</a></li>
    </ul>
  </nav>
</header>
```

Рисунок 5 – Код навигационной панели

Итоговой вариант навигационной панели представлен на рисунке 6

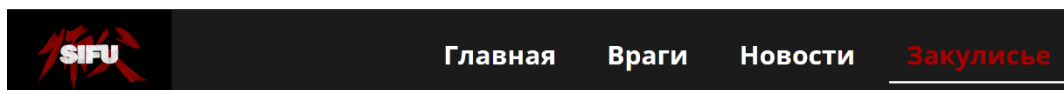


Рисунок 6 – Навигационная панель

2. Web-сценарии сайта на языке JavaScript

2.1 Назначение языка JavaScript

Помимо ядра JavaScript, в веб-программировании также широко используются различные библиотеки и фреймворки на основе JavaScript. Они предоставляют готовые решения и инструменты, упрощающие разработку и расширение функциональности веб-сайтов.

В разработанном мной сайте игры SIFU я также использовал JavaScript и несколько библиотек (Express, Intersection Observer API). Вот некоторые из их назначений:

1. Рутинг: Я использовал JavaScript-библиотеку Express для рутинга, которая помогает управлять навигацией на сайте. Она позволяет облегчить написание логики переключения между различными страницами
2. Шаблонизация: Я использовал JavaScript-библиотеку для шаблонизации, которая позволяет генерировать динамический контент на веб-страницах. Это упрощает создание и обновление различных элементов, таких как список новостей или информация о врагах, с использованием шаблонов, в которых можно заполнять данные из источников, таких как база данных или API.
3. Деплой на локальной сети: Для демонстрации и тестирования сайта на локальной сети я использовал платформу Node JS, которая позволяет развернуть веб-сервер и делать сайт доступным для других устройств внутри той же сети. Это позволяет мне проверить функциональность и внешний вид сайта.

2.2 Разработка web-сценариев для сайта

В разработанном сайте игры SIFU были реализованы следующие веб-сценарии с целью обеспечения интерактивности и функциональности:

Переключение между страницами: для обеспечения плавной загрузки контента был реализован сценарий переключения между различными страницами с помощью роутинга. Пример кода с комментариями представлен на рисунке 7

```
//Главная страница
app.get('/', function (request, response) {
  const output = tempMain;
  response.send(output);
});

//Новости
app.get('/news', function (request, response) {
  const cardsHtml = dataNewsObj.map((el) => replaceNewsTemplate(tempNewsCard, el)).join(''); //цикл
  const output = tempNews.replace('%NEWS%', cardsHtml);
  response.send(output);
});

//Страница с новостью
app.get('/news-page', function (request, response) {
  const product = dataNewsObj[request.query['id']];

  const output = replaceNewsTemplate(tempNewsPage, product);
  response.send(output);
});
```

Рисунок 7 – Код роутинга

Замена данных из JSON в шаблон на веб-странице: был написать JavaScript-модуль, который будет загружать данные из JSON-файла, а затем заполнять шаблон соответствующими значениями. Пример кода для такого модуля с комментариями представлен на рисунке 8

```
module.exports = (temp, news) => {  
  ...  
  let output = temp.replace(/{%IMG%}/g, news.img);  
  output = output.replace(/{%DATE%}/g, news.date);  
  output = output.replace(/{%TITLE%}/g, news.title);  
  output = output.replace(/{%DESCRIPTION%}/g, news.description);  
  output = output.replace(/{%NEWS-TEXT%}/g, news.news_text);  
  output = output.replace(/{%ID%}/g, news.id);  
  
  return output;  
};
```

Рисунок 8 – Код модуля

3. Валидация кода и продвижение сайта.

3.1 Валидация кода сайта

Для обеспечения качества и соответствия стандартам, была проведена проверка кода HTML и CSS на валидность. Результаты проверки кода представлены ниже:

1. Валидация кода HTML:

- Результат проверки: HTML код прошел валидацию и не содержит ошибок.
- Замечания: Ошибки отсутствуют.

2. Валидация кода CSS:

- Результат проверки: CSS код прошел валидацию и не содержит ошибок.
- Замечания: Ошибки отсутствуют.

В результате проверки кода HTML и CSS не было выявлено никаких замечаний или ошибок. Это свидетельствует о том, что код соответствует стандартам и рекомендациям, что является хорошей практикой разработки веб-сайтов.

Примечание: В случае, если были обнаружены замечания или ошибки в коде, рекомендуется проанализировать каждое замечание и внести соответствующие исправления. Замечания могут касаться неправильного синтаксиса, неподдерживаемых атрибутов или свойств, отсутствия обязательных элементов и других подобных проблем.

Для проверки был использован плагин Prettier

3.2 Способы продвижения сайтов

Существует множество способов продвижения сайтов с целью повышения их позиции в поисковых системах. Некоторые из наиболее известных и эффективных приемов включают:

1. Оптимизация поисковой системы (SEO): это комплекс мер, направленных на улучшение видимости и ранжирования сайта в органических результатах поиска. Он включает в себя оптимизацию контента, использование ключевых слов, создание качественных обратных ссылок, оптимизацию заголовков и мета-тегов, улучшение скорости загрузки страницы и другие факторы, которые способствуют улучшению ранжирования сайта в поисковых системах.
2. Оптимизация скорости загрузки страницы: быстрая загрузка страницы важна для удобства пользователей и для ранжирования в поисковых системах. Оптимизация изображений, использование кэширования, сжатие файлов и другие методы помогают ускорить время загрузки страницы
3. Создание качественного контента: создание уникального, информативного и привлекательного контента является одним из ключевых факторов, влияющих на ранжирование сайта. Хороший контент привлекает посетителей, увеличивает время их пребывания на сайте, способствует разделению контента в социальных сетях и получению качественных обратных ссылок

3.3 Используемые приемы продвижения сайта

В данной работе были использованы следующие приемы продвижения сайта:

1. Оптимизация поисковой системы (SEO):

Внедрение метатегов: для каждой страницы сайта были определены метатеги, такие как **<title>**, **<meta description>**, **<meta keywords>**. Они содержат релевантные ключевые слова и описание страницы, которые помогают поисковым системам понять контекст и содержание страницы. Пример кода с метаданным представлен на рисунке 9

```
<html lang="ru">
<head>
  <title>SIFU</title>
  <meta name="description" content="Главная страница" />
  <meta name="keywords" content="боевые искусства, игры, сифу" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Рисунок 9 – Оптимизация поисковой системы

2. Оптимизация скорости загрузки страницы.

Была проведена оптимизация изображений. Сайт использует кэширование

4. Итоговой вариант web-сайта

Ниже приведен итоговый вариант веб-сайта, разработанного в рамках данной курсовой работы. Представлены скриншоты основных страниц сайта, чтобы дать представление о его внешнем виде и организации контента.

На главной странице представлен общий обзор игры SIFU, а также важные новости и актуальная информация о проекте. Результат на рисунках 10, 11, 12

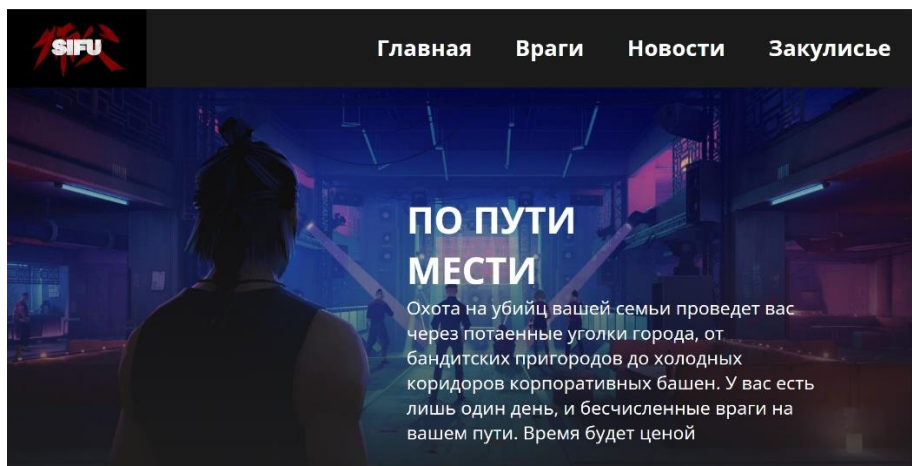


Рисунок 10 – Главная страница (1)

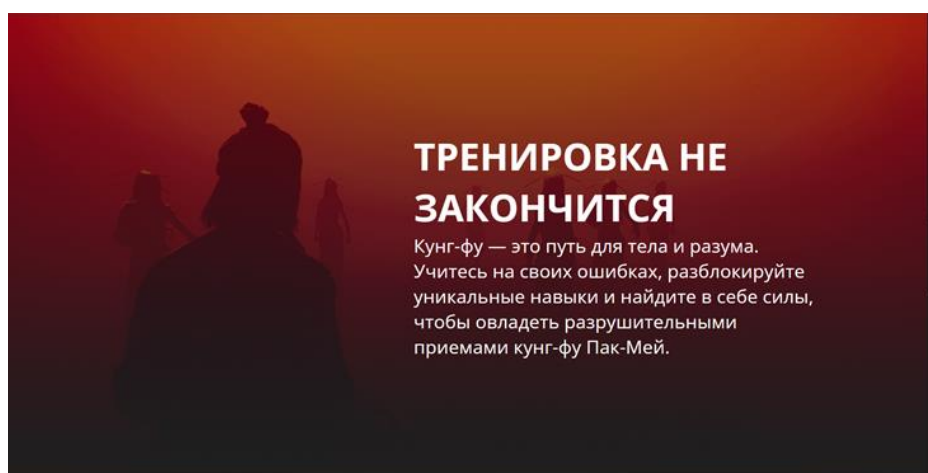


Рисунок 11 – Главная страница (2)

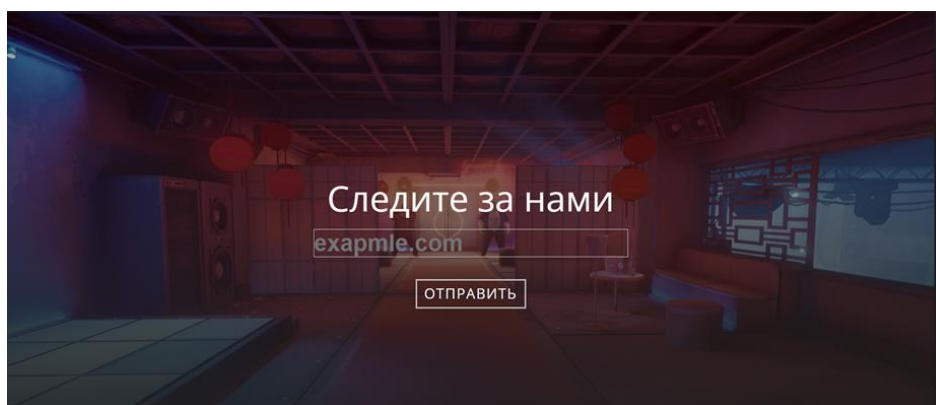


Рисунок 12 – Главная страница (3)

На странице "Враги" представлены основные персонажи, с которыми игрок сталкивается в игре. Каждый персонаж сопровождается кратким описанием и иллюстрацией. Также есть возможность перейти на отдельную страницу с тактикой ведения боя. Результат на рисунках 13, 14

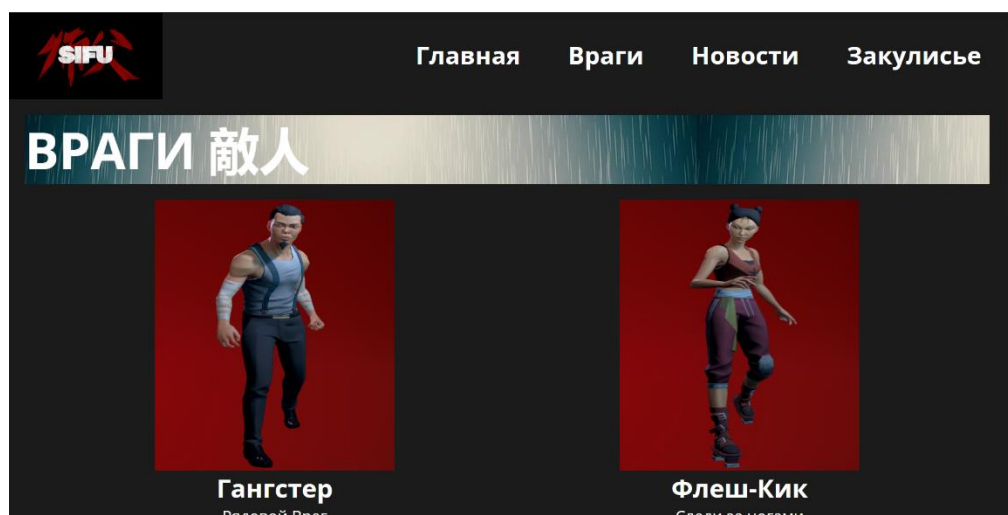


Рисунок 13 – Страница Враги

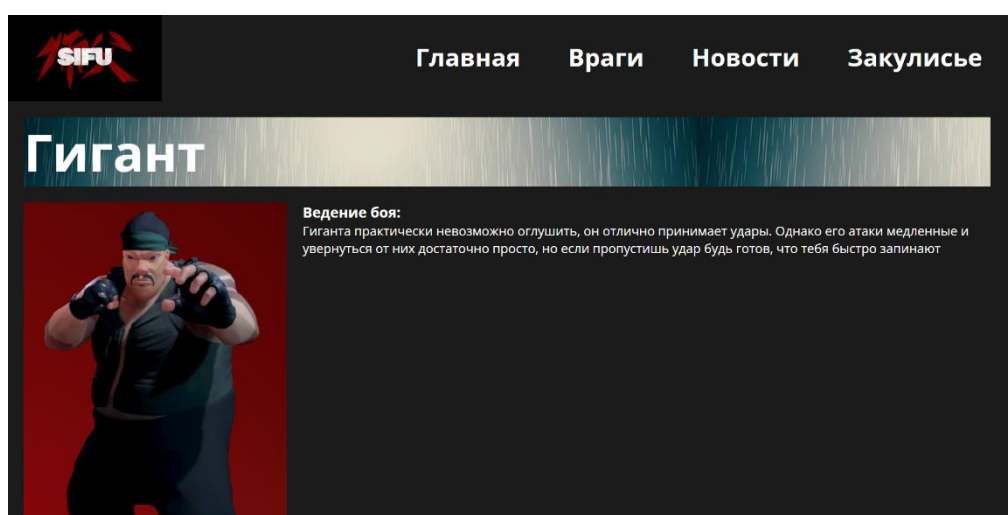


Рисунок 14 – Отдельная страница врага

На странице "Новости" пользователи могут ознакомиться с последними новостями и событиями, связанными с игрой SIFU. Результат на рисунках 15, 16, 17

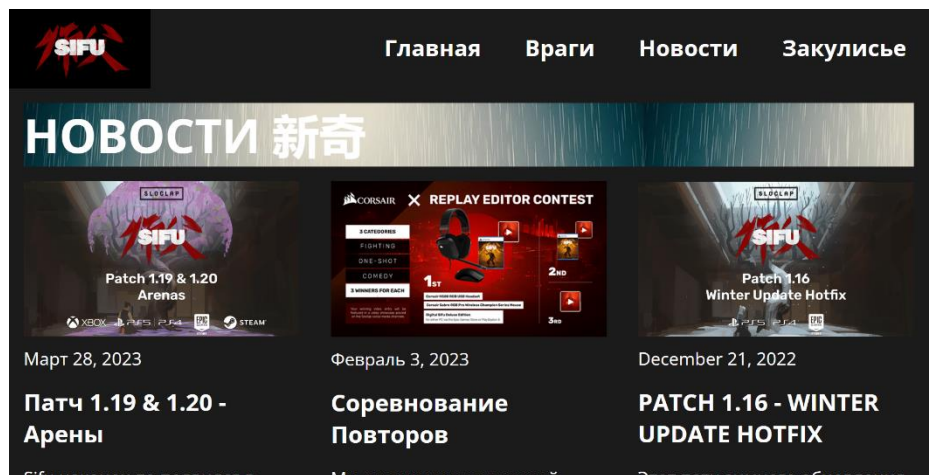


Рисунок 15 – Страница Новости (1)

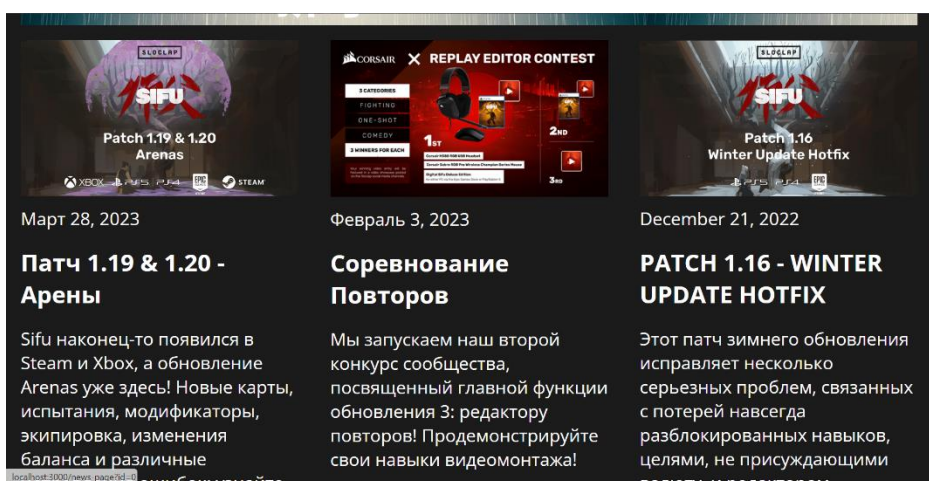


Рисунок 16 – Страница Новости (2)

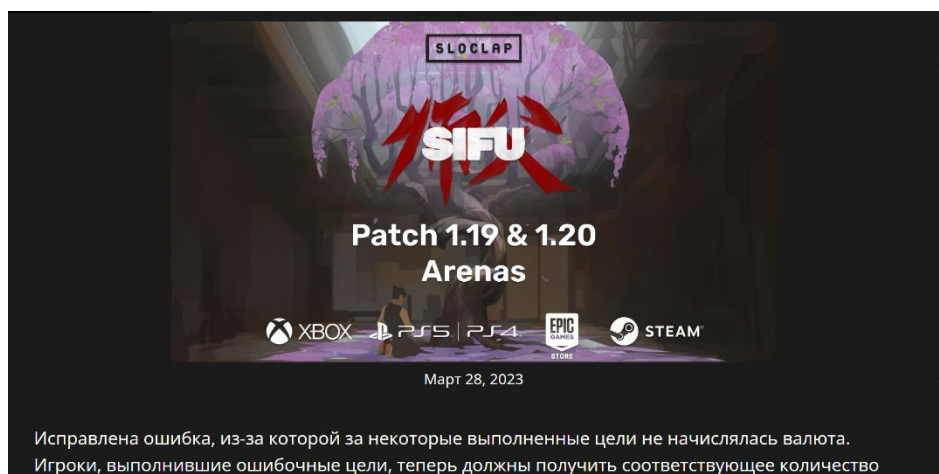


Рисунок 17 – Отдельная страница новости

Страница "Закулисье" предлагает интересные факты о разработке игры, интервью с разработчиками. Результат на рисунках 18, 19



Рисунок 18 – Страница Закулисье (1)



Рисунок 19 – Страница Закулисье (2)

На рисунках 20, 21 представлен пример адаптивности с использованием медиа-запросов для страницы Новости

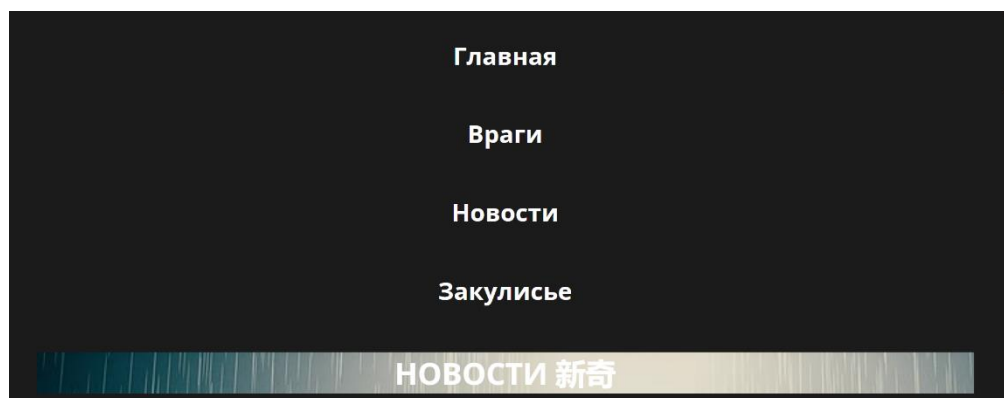


Рисунок 20 – Адаптивность (1)

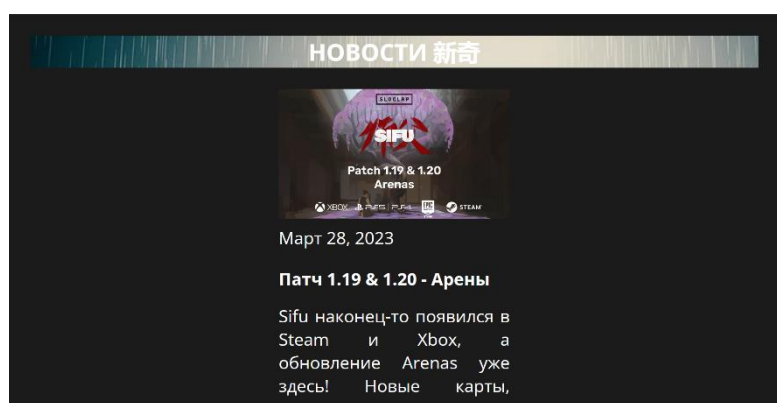


Рисунок 21 – Адаптивность (2)

Заключение

В заключении хочется отметить, что данная курсовая работа была выполнена с целью закрепления навыков создания веб-сайта и использования современных технологий веб-дизайна. В ходе работы был разработан тематический сайт по игре SIFU, который включает в себя различные разделы, такие как "Враги", "Новости", "Закулисье"

Была проведена работа над дизайном сайта, создан макет с использованием приложения Figma, а также реализована адаптивная верстка с применением технологий HTML и CSS. Также была применена технология JavaScript для реализации функциональности, такой как рутинг, шаблонизация и деплой на локальной сети.

Для продвижения сайта были использованы приемы оптимизации поисковой системы (SEO). Были также рассмотрены вопросы валидации кода и оптимизации графики сайта.

В целом, курсовая работа достаточно полно реализовала поставленные задачи. Сайт готов для предполагаемой публикации в Интернете, однако для полноценного использования рекомендуется провести дополнительное тестирование, оптимизацию и адаптацию под различные браузеры и устройства.

Работа позволила закрепить и применить полученные знания в области создания веб-сайтов и использования современных технологий веб-дизайна.

Список использованных источников

1. Красильникова О.И., Красильников Н.Н. Web-технологии для разработки клиентской части web-страниц: учеб. пособие. Ч 1 – СПб.: ГУАП, 2017 – 59 с.
2. Красильникова О.И., Красильников Н.Н. Web-технологии для разработки клиентской части web-страниц: учеб. пособие в 2 ч. Ч. 2 – СПб.: ГУАП. 2018 - 44 с.
3. Красильникова, О. И. Технологии верстки в web-дизайне: учеб. Пособие / О. И. Красильникова. – СПб.: ГУАП, 2020 – 82 с.
4. Красильникова, О. И. JavaScript в разработке клиентской части веб-страниц: учеб. пособие / О. И. Красильникова. – СПб.: ГУАП, 2022 – 87 с.
5. Государев И. Б. Введение в веб-разработку на языке JavaScript: учебное пособие / И. Б. Государев. СПб: Лань, 2019 144 с.

Приложение А

App.js

```
const express = require('express');
const fs = require('fs');
const replaceNewsTemplate = require('./modules/replace_news_template');

const app = express();

// Шаблоны
const tempMain = fs.readFileSync(`${__dirname}/templates/template-main.html`,
'utf-8');
const tempBhScene = fs.readFileSync(`${__dirname}/templates/template-behind-
scene.html`, 'utf-8');

const tempNews = fs.readFileSync(`${__dirname}/templates/template-news.html`,
'utf-8');
const tempNewsCard = fs.readFileSync(`${__dirname}/templates/template-news-
card.html`, 'utf-8');
const tempNewsPage = fs.readFileSync(`${__dirname}/templates/template-news-
page.html`, 'utf-8');

const tempEnemies = fs.readFileSync(`${__dirname}/templates/template-
enemies.html`, 'utf-8');
const tempEnemiesCard = fs.readFileSync(`${__dirname}/templates/template-
enemies-card.html`, 'utf-8');
const tempEnemiesPage = fs.readFileSync(`${__dirname}/templates/template-
enemies-page.html`, 'utf-8');

const dataNews = fs.readFileSync(`${__dirname}/data/news_Data.json`, 'utf-
8');
const dataNewsObj = JSON.parse(dataNews); //Array of product objects

const dataEnemies = fs.readFileSync(`${__dirname}/data/enemy_Data.json`,
'utf-8');
const dataEnemiesObj = JSON.parse(dataEnemies); //Array of product objects

// Подключаем источник всех изображение, стилей, скриптов
app.use(express.static(__dirname + '/public'));

//Главная страница
app.get('/', function (request, response) {
  const output = tempMain;
  response.send(output);
});

//Новости
```



```

app.get('/news', function (request, response) {
  const cardsHtml = dataNewsObj.map((el) => replaceNewsTemplate(tempNewsCard,
el)).join(''); //цикл
  const output = tempNews.replace('{%NEWS%}', cardsHtml);
  response.send(output);
});

//Страница с новостью
app.get('/news-page', function (request, response) {
  const product = dataNewsObj[request.query['id']];

  const output = replaceNewsTemplate(tempNewsPage, product);
  response.send(output);
});

//Закулисье
app.get('/behind_scene', function (request, response) {
  const output = tempBhScene;
  response.send(output);
});

//Враги
app.get('/enemies', function (request, response) {
  const cardsHtml = dataEnemiesObj.map((el) =>
replaceNewsTemplate(tempEnemiesCard, el)).join(''); //loop
  const output = tempEnemis.replace('{%CARDS%}', cardsHtml);
  response.send(output);
});

//Страница с врагом
app.get('/enemies-page', function (request, response) {
  const product = dataEnemiesObj[request.query['id']];

  const output = replaceNewsTemplate(tempEnemiesPage, product);
  response.send(output);
});

app.listen(3000);

```

enemy_Data.json

```

[
  {
    "id": 0,
    "title": "Гангстер",
    "description": "Рядовой Враг",
    "img": "Gangster.png",

```

```

    "news_text": "Самый слабый и самый предсказуемый враг. Умеет блокировать атаки, однако от всех его атак можно уклониться и контратаковать. Играйте от защиты"
  },
  {
    "id": 1,
    "title": "Флеш-Кик",
    "description": "Следи за ногами",
    "img": "Flash-Kick.png",
    "news_text": "Большинство атак этой барышни приходится на различные вариации хай-киков. Тем не менее руками она бьет тоже и после удара рукой использует подсечку"
  },
  {
    "id": 2,
    "title": "Гигант",
    "description": "Двойные проблемы",
    "img": "Gigant.png",
    "news_text": "Гиганта практически невозможно оглушить, он отлично принимает удары. Однако его атаки медленные и увернуться от них достаточно просто, но если пропустишь удар будь готов, что тебя быстро запинаят"
  }
]

```

template-enemies.html

```

<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
  href="https://fonts.googleapis.com/css2?family=Lobster&family=Open+Sans:wght@400;700&display=swap"
  rel="stylesheet"
/>
<link rel="stylesheet" href="/css/general.css" />
<link rel="stylesheet" href="/css/enemies_card.css" />
<link rel="stylesheet" href="/css/enemies.css" />
<link rel="stylesheet" href="/css/content.css" />
<link rel="stylesheet" href="/css/flips.css" />

<header>
  <div class="logo">
    
  </div>
  <nav>
    <ul>
      <li><a class="page-link" href="http://localhost:3000">Главная</a></li>
      <li><a class="page-link"
href="http://localhost:3000/enemies">Враги</a></li>

```

```

        <li><a class="page-link"
href="http://localhost:3000/news">Новости</a></li>
        <li><a class="page-link"
href="http://localhost:3000/behind_scene">Закулисье</a></li>
    </ul>
</nav>
</header>

<body>
    <div class="content">
        <div class="content-header" style="background-image:
url('/img/Contents/news.jpg')">
            <div class="scroll-reveal box left">ВРАГИ 敵人</div>
        </div>
    </div>
    <div class="cards-box">{%CARDS%}</div>
    <script src="/js/flips.js"></script>
</body>

```

template-enemies-card.html

```

<figure class="card">
    

    <div class="card-text">
        <div class="card-title-box">
            <h2 class="card-title">{%TITLE%}</h2>
        </div>

        <div class="card-detail-box">
            <p class="card-detail">{%DESCRIPTION%}</p>
        </div>

        <a class="card-link" href="/enemies-page?id={%ID%}">
            <span class="read-more">ПРОЧИТАТЬ</span>
        </a>
    </div>
</figure>

```

enemies_card.css

```

a {
    color: white;
    text-decoration: none;
}

.image-link {
    min-width: 250px;
    min-height: 250px;
}

```

```

padding: 13px;
display: inline-block;
}

.read-more {
  font-size: 28px;
  font-weight: bolder;
}

.cards-box {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-evenly;
}

.card {
  width: 40%;

  height: auto;
  display: flex;
  flex-direction: column;

  justify-content: flex-start;

  align-items: center;
}

.card-text {
  text-align: center;
}

.card-date {
  margin-top: 3%;
  margin-bottom: 10px;
  font-size: 24px;
  font-weight: 50;
}

.card-title {
  font-size: 32px;
  font-weight: bolder;
}

.card-detail-box {
  max-height: 38%;
  display: block;
  overflow: hidden;
}

```

```

.card-detail {
  font-size: 20px;
}

.card-image {
  width: 60%;
  height: 55%;
  vertical-align: middle;
  display: inline-block;
  cursor: pointer;
  margin: 0 auto;
}

.card-link {
  padding: 0;
  margin-top: 15px;
}

.card-link:hover {
  border-bottom: 3px solid #fff;
  transition: all 0.3s ease-in-out;
}

.card-link:hover::after {
  width: 100%;
}

@media screen and (max-width: 800px) {
  .cards-box {
    display: inline;
  }

  .card {
    text-align: justify;
    margin-left: auto;
    margin-right: auto;
  }

  .card-date {
    font-size: 1em;
  }

  .card-title {
    font-size: 1em;
  }

  .card-detail {

```

```
    font-size: 0.7em;
  }

  .card-link {
    text-align: center;
    margin-top: 5px;
    margin-bottom: 10px;
  }

  .read-more {
    font-size: 1em;
  }

  .image-link {
    all: unset;
  }

  .card-image {
    margin: 0 auto;
  }
}
```