**1 Анализ задачи**

* 1. **Постановка задачи**

На учебную практику по WEB-программированию была поставлена задача по вёрстке шаблона по макету из Figma на тему: Сайт магазина компании Instroom.

Требуется сверстать главную страницу сайта по макету, представленному в виде figma-ссылки:

https://www.figma.com/file/4UwtLyhuq8uvxRxhGrVc7R/ИМ-стройматериалов?type=design&node-id=0-1&t=rqMwhZhpCqtFpvh6-0

Главная страница должна быть сверстана с учётом просмотра её на различных устройствах с разными разрешениями экрана (ширина по умолчанию – 1240px, макет соответствует ширине 800px).

Также должен быть составлен документ, который содержит в себе список элементов и правил, подходящих под стилистику конкретного сайта – стайлгайд. Стайлгайд – это справка с набором стандартов и требований, обязательных к соблюдению. Их необходимо использовать в процессе оформления сайта. Ключевое назначение – создание единого стилистического и оформительского однообразия. На основании данных из Figma стайлгайд должен включать в себя следующие разделы:

* Логотип;
* Цветовая палитра;
* Типографика;
* Сетка и отступы;
* Навигация, меню;
* Кнопки, поля ввода;
* Компоненты;

После завершения составления стайлгайда и вёрстки макета необходимо выполнить стилизацию с помощью CSS отдельных элементов макета в соответствии со следующими критериями:

* Hover-эффекты;
* Оформление гиперссылок;
* Оформление изображений;
* Декоративные элементы;
* Любые другие эффекты;

Далее необходимо добавить в сверстанный шаблон главной страницы интерактивные, динамические элементы. При их разработке необходимо писать код на JavaScript и использовать библиотеку Jquery, Jquery UI в соответствии со следующими критериями:

* Работа с окнами сообщений;
* Создание динамического меню в организации навигации;
* Использование любых возможных способов JS+библиотеки.

**1.2 Инструменты разработки**

Для вёрстки данного шаблона макета будет выбрана среда Microsoft Visual Studio Code со следующими установленными расширениями: Live Server А также для дальнейшей работы с проектом потребуется Figma и GitHub.

Microsoft Visual Studio Code – редактор исходного кода. Позиционируется как «легкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

При верстке данного шаблона макета будет использован персональный компьютер со следующими характеристиками:

– Процессор Intel(R) core i3 5005h

– ОЗУ: 8Гб;

* Память HDD 1024Гб;
* ОС Windows 10.

**2. Проектирование задачи**

**2.1 Стайлгайд**

**2.1.1 Логотип и сетка**

Логотип — графический знак, эмблема или символ, используемый территориальными образованиями, коммерческими предприятиями, организациями и частными лицами для повышения узнаваемости и распознаваемости в социуме. Логотип представляет собой название сущности, которую он идентифицирует, в виде стилизованных букв и/или идеограммы. (Рисунок 1 — Логотип)



Рисунок 1 — Логотип

Опишем главные «Не» для верстки логотипа данного сайта:

1. Нельзя делать перенос слов;
2. Не стоит модернизировать логотип;
3. Нельзя менять расстояние между буквами;
4. Нельзя менять шрифт;

Модульная сетка сайта для персонального компьютера (ширина 1920px), представлена в приложении А.

**2.1.2 Цветовая палитра**

Таблица 1 — Цветовая таблица сайта

|  |  |
| --- | --- |
| Элемент | Цвет |
| 1 | 2 |
| Основные цвета сайта | #F2F5F7(Белый) |
| #212526(Серый) |
| #F05A00(Оранжевый) |
| #000000(Черынй) |

**2.1.2 Типографика**

Типографика — сильнейший инструмент для выражения посыла в веб-дизайне. С его помощью вы можете объединить текстовую и визуальную составляющие, что поможет вам достучаться до посетителя. Типографика для шаблона макета «» представлена в таблице 2.

|  |  |  |
| --- | --- | --- |
| Элемент | Шрифт | Размер |
| 1 | 2 | 3 |
| Логотип (текст) | Prosto One | 24px |
| Текст рядом с логотипом | Raleway | 16px |
| Текст в форме | Raleway | 16px |
| Заголовок раздела страницы | Prosto One | 30px |
| Всплывающий текст на картинке | Prosto One | 40рх |
| Текст в слайдере | Raleway | 16рх |
| Названия разделов footer-а | Prosto One | 30рх |

**2.1.4 Навигация и меню**

Важнейшей составляющей любого сайта является навигация и меню. Частая ошибка юзабилити-сайта — это сложная или запутанная навигация, которая отталкивает клиента. Простое правило: чем проще ориентироваться на сайте, тем быстрее пользователь найдёт необходимую ему информацию.

В шапке сайта пользователь без трудностей может перейти к различным разделам сайта. (Рисунок 2)

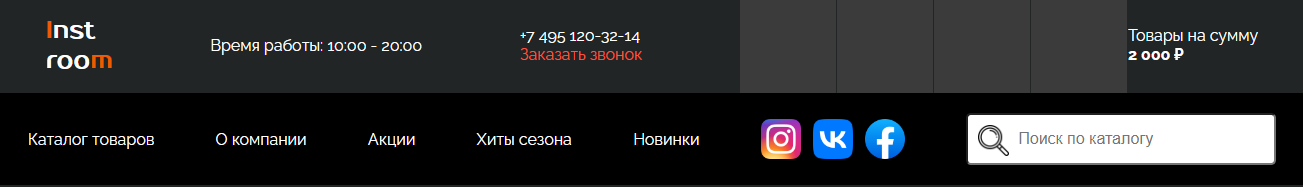


Рисунок 2 — Шапка сайта

Подвал данного макета имеет полезную информацию о гостевом доме.(Рисунок 3)

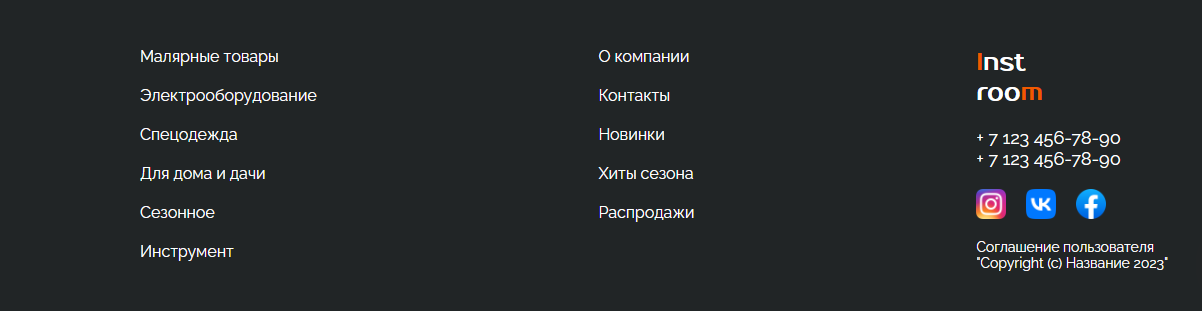


Рисунок 3 — Подвал сайта

С помощью кнопки «Наверх» пользователь может подняться в начало сайта.

**2.1.5 Иконки**

Иконка — это максимально упрощенное изображение объекта или действия. Иконки идеально вписываются в современные тренды веб-дизайна. Помимо этого, у иконок есть ещё одно неоспоримое преимущество перед текстом — отсутствие языкового барьера. Иллюстрации понимают все, поэтому иконки используют в операционных системах, интерфейсах сайта, в презентациях и в рекламе.

Элементы шапки сайта (Рисунок 4).



Рисунок 4 — Иконки шапки

**2.1.6 Кнопки и поля ввода**

Интерактивные элементы пользовательского интерфейса создают основу мощных и удобных цифровых продуктов. Внимательный подход к деталям помогает создать положительный опыт взаимодействия с продуктом в целом. Кнопка является одним из самых распространённым элементом любого интерфейса.

Форма для проверки свободных комнат в доме (Рисунок 6)

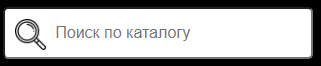


Рисунок 5 — кнопка в поле поиск

Рисунок 6 — стрелка для переключения слайдера

**2.1.7 Компоненты**

Кнопка «Наверх» (Рисунок 6)



Рисунок 6 — Кнопка «наверх»

**3 Реализация**

**3.1 Руководство программиста**

**3.1.1 Общие сведения**

Программный продукт реализован с помощью библиотеки тегов HTML и каскадной таблицы стилей CSS/ В сверстанный шаблон были добавлены динамические элементы на CSS. Добавлены в сверстанный шаблон главной страницы интерактивные, динамические элементы. При их разработке использовался язык Java Script.

Для реализации адаптивной верстки использовалась технология flexbox, а также медиа-запросы. В главном теге body имеются необходимые разделы header, main и footer

Шапка сайта заключена в контейнер header

Этот контейнер имеет следующие свойства:

<header>

<p class="header-logo">

<span style="color: #F05A00">I</span>nst<br>

roo<span style="color: #F05A00">m</span></p>

<p>Время работы: 10:00 - 20:00</p>

<p>+7 495 120-32-14<br><a href="">Заказать звонок</a></p>

<div class="header-blocks">

<div class="header-item">

</div>

<div class="header-item">

</div>

<div class="header-item">

</div>

<div class="header-item">

</div>

<p>Товары на сумму<br><b>2 000 ₽</b></p>

</div>

</header>

<div class="header2">

<p>Каталог товаров</p>

<p>О компании</p>

<p>Акции</p>

<p>Хиты сезона</p>

<p>Новинки</p>

<div class="header2-icon">

<img src="pic/inst.png" width="40px">

<img src="pic/vk.png" width="40px">

<img src="pic/Facebook.png" width="40px">

</div>

<form>

<input type="text" placeholder="Поиск по каталогу">

</form>

</div>

Внутри первого контейнера находится контактная информация, внутри второго находится логотип и меню сайта, а также иконки социальных сетей. Данные контейнеры имеют следующие стили:

header{

width: 100%;

background-color:#212526;

display: flex;

justify-content: space-around;

align-items: center;

}

header p{

color: white;

font-size: 16px;

}

.header-logo{

font-family: 'Prosto One';

font-size: 24px;

}

header a{

color: #FC573B;

}

.header-blocks{

display: flex;

gap: 1px;

align-items: center;

}

.header-item{

width: 96px;

height: 96px;

background-color: #3B3B3B;

}

.header2{

width: 100%;

background-color:black;

padding: 20px 0px 20px 0px;

display: flex;

justify-content: space-around;

align-items: center;

}

.header2 p{

color: white;

}

.header2-icon{

display: flex;

gap: 12px;

}

.header2 form input{

width: 310px;

height: 52px;

padding: 0px 0px 0px 50px;

box-sizing: border-box;

border-radius: 5px;

font-size: 16px;

background-image: url(/pic/lopa.png);

background-size: 10%;

background-repeat: no-repeat;

background-position:10px 10px;

}

Раздел main включает в себя контейнер с брендами

<div class="brends">

<p>Наши бренды</p>

<img src="truper.png">

<img src="центро.png">

<img src="finland.png">

<img src="Kapro.png">

<img src="truper.png">

<img src="центро.png">

</div>

Имеющий следующие стили:

.brends{

width: 100%;

height: 246px ;

background-color:#212526;

display: flex;

justify-content: space-around;

align-items: center;

color: white;

}

.brends p{

position: absolute;

width: 234px;

height: 32px;

left: 11px;

top:760px;

font-family: 'prosto one';

font-size: 20px;

}

Далее идет блок, в котором находятся новости

<div class="news-items">

<div class="news-items-column">

<div class="news-item">

<img src="pic/news-img.png" title="Новость" alt="Новость">

<h4>Название новости</h4>

<p>12 января 2023</p>

</div>

<div class="news-item">

<img src="Rectangle 49.png" title="Новость" alt="Новость">

<h4>Название новости</h4>

<p>12 января 2023</p>

</div>

</div>

<div class="news-items-column">

<div class="news-item">

<img src="Rectangle 49.png" title="Новость" alt="Новость">

<h4>Название новости</h4>

<p>12 января 2023</p>

</div>

<div class="news-item">

<img src="pic/news-img.png" title="Новость" alt="Новость">

<h4>Название новости</h4>

<p>12 января 2023</p>

</div>

</div>

</div>

Данный блок имеет следующие стили:

#news {

display: flex;

justify-content: center;

padding: 50px 0 50px 0;

background-color: #f2f5f7;

width: 100%;

}

.news-container {

display: flex;

gap: 40px;

}

.news-text {

display: flex;

flex-direction: column;

gap: 30px;

}

.news-text h3 {

font-family: 'Prosto One';

font-size: 30px;

color: black;

}

.news-text p {

font-size: 16px;

line-height: 25px;

}

.news-text form button {

width: 250px;

height: 50px;

background-color: #F05A00;

border-radius: 5px;

border: none;

color: white;

font-size: 16px;

}

.news-text form button:hover {

background-color: #3b3b3b;

}

.news-items {

display: flex;

gap: 10px;

}

.news-items-column {

display: flex;

flex-direction: column;

gap: 30px;

}

.news-item {

display: flex;

flex-direction: column;

gap: 10px;

}

Далее идет раздел footer в котором находится подвал сайта, с контактами, полезной информацией

<footer>

…

</footer>

**3.1.2 Динамические элементы на CSS**

На странице сайта реализованы такие динамические элементы, как:

– HOVER-эффекты к кнопкам

– Использование теней к кнопкам и некоторым блокам

– Всплывание информации при наведении на картинку

**3.1.3 Элементы на JavaScript**

На странице были реализованы такие элементы как:

– Кнопка «Наверх»

var scrollToTopButton = document.getElementById("scrollToTopButton");

window.addEventListener("scroll", function () {

if (window.pageYOffset > 200) {

// Если прокручена более 100 пикселей, показать кнопку

scrollToTopButton.classList.add("show");

} else {

// В противном случае скрыть кнопку

scrollToTopButton.classList.remove("show");

}

});

scrollToTopButton.addEventListener("click", function () {

// Плавно прокрутить страницу наверх

window.scrollTo({

top: 0,

behavior: "smooth"

});

});

– Проверка куки

function setCookie(name, value, days) {

let expires = "";

if (days) {

let date = new Date();

date.setTime(date.getTime() + (days \* 24 \* 60 \* 60 \* 1000));

expires = "; expires=" + date.toUTCString();

}

document.cookie = name + "=" + (value || "") + expires + "; path=/";

}

function getCookie(name) {

let matches = document.cookie.match(new RegExp("(?:^|; )" + name.replace(/([\.$?\*|{}\(\)\[\]\\\/\+^])/g,

'\\$1') + "=([^;]\*)"));

return matches ? decodeURIComponent(matches[1]) : undefined;

}

let cookieNote = document.getElementById('cookie-note');

let cookieBtnAccept = cookieNote.querySelector('.cookie\_\_accept');

// Если куки cookies\_policy нет или она просрочена, то показываем уведомление

if (!getCookie('cookies\_policy')) {

cookieNote.classList.add('show');

}

// При клике на кнопку устанавливаем куку cookies\_policy на 3 дня

cookieBtnAccept.addEventListener('click', function () {

setCookie('cookies\_policy', 'true', 3);

cookieNote.classList.remove('show');

});

**3.2 Спецификация файлов проекта**

Спецификация файлов представлена в таблице 2.

Таблица 2 – Спецификация файлов.

|  |  |  |
| --- | --- | --- |
| Файл | Название | Расширение |
| Файлы с HTML кодом | index | .html |
| Файлы с CSS кодом | main | .css |
| Файлы с JS кодом | button-up cookie | .js |

Также в общей папке имеется папка img, в которой хранятся все картинки и иконки разных форматов (.svg, .png)

**4 Тестирование**

Тестирование является одним из самых важных этапов при создании программного продукта. Исходя из последующего внедрения программы в использование, необходимо осуществить тестирование по набору тестов, для того чтобы убедиться в корректной работе ПП. Опираясь на данный набор тестов, можно говорить о некоторой степени уверенности в правильности работы программного продукта.

Таким образом основная часть ошибок и недоработок была выявлена и исправлена на этапе реализации проекта. После завершения этапа реализации, было проведено тщательное тестирование.

Программный продукт был протестирован разработчиком Крыгин Алексей Константинович. Расписание проведения и время, затраченное на тестирование, описано в таблице 3.

Таблица 3 – Расписание проведения тестирования.

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Дата | Описание | Длительность, ч |
| Крыгин А.К. | 23.05.2023 | Выполнение тест-кейсов | 3 |
| Крыгин А.К. | 24.05.2023 | Анализ выполнения тест-кейсов | 2 |
| Крыгин А.К. | 24.05.2023 | Повторное выполнение тест-кейсов | 2 |

Выявленные по результатам ошибки тестирования приводятся в таблице 3.

Таблица 4 – Статистика по выявленным ошибок.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Статус | Количество ошибок | Важность | | | |
| Критическая | Высокая | Средняя | Минимальная |
| Исправлено | 3 | 1 | 0 | 0 | 2 |
| Проверено | 3 | 0 | 0 | 0 | 0 |
| Открыто заново | 0 | 0 | 0 | 0 | 0 |
| Найдено | 3 | 0 | 0 | 0 | 0 |
| Отклонено | 0 | 0 | 0 | 0 | 0 |

Таким образом, после проведения тестирования, критических ошибок выявлено не было, однако были найдены ошибки, связанные с подключением шрифтов из папки со шрифтами, ошибки с подключением картинок.

Опираясь на вышеперечисленное, можно сделать вывод о стабильности работы программного продукта и его основного функционала. Все ошибки, влияющие на работу сайта, были устранена, однако ошибки низкой важности остались, исходя из чего можно сделать вывод, что тестирование прошло успешно.

**5 Применение**

**5.1 Размещение на GitHub**

Данный учебный проект размещается на GitHub в репозитории по ссылке:

https://uliveee.github.io/InstRoom/

**Заключение**

Целью учебной практики являлась вёрстка шаблона макета из Figma.

В ходе реализации поставленной задачи был укреплён пройденный изученный материал по дисциплине «Программные средства создания Internet-приложений», а также получено множество информации о возможностях вёрстки.

После тщательного тестирования сайта были выявлены некоторые недоработки, которые были исправлены на стадии проектирования. В целом при реализации программы, были выполнены все необходимые условия. Таким образом, можно сказать, что программа была реализована успешно.

**Список использованных источников**

<http://htmlbook.ru/> — справочник по HTML и CSS

**Приложение А**

**Модульная сетка сайта**

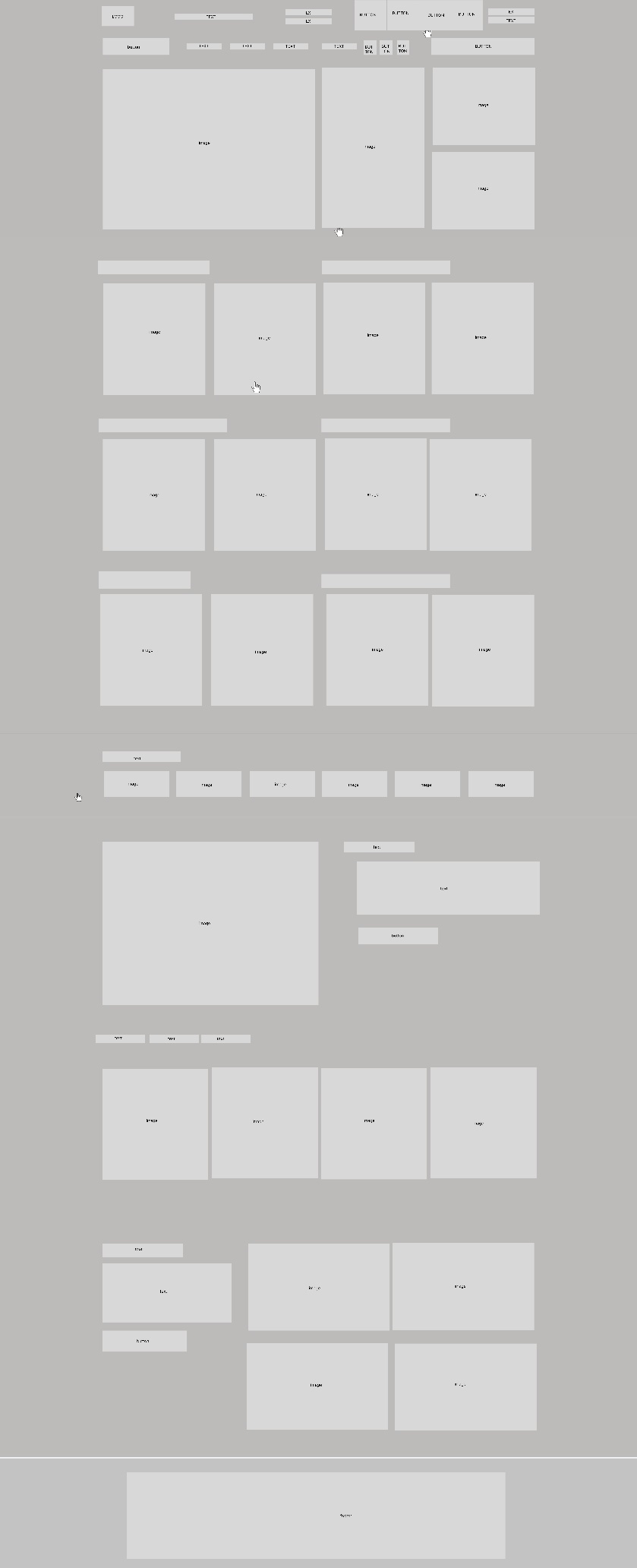


Рисунок 7 - Макет ПК версии сайта