**Автономное профессиональное образовательное учреждение Вологодской области**

**«Вологодский колледж связи и информационных технологий»**

**Отчёт**

О прохождении практики (учебная, производственная);

(Вид практики)

Студентом Кравченко Ильей Александровичем

Специальность «Информационные системы и программирование»

Курс 4 Форма обучения ОЧНАЯ Группа ИСП 421р

В Студии интернет-решений «Grampus»

(наименование организации)

тема

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

с «14» октября 2024г. по «29» ноября 2024г.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2024г.

Подпись руководителя

практики от колледжа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Наталья Вениаминовна Зернова

(подпись) (расшифровка)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2024г.

Отчет принял

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Роман Андреевич Дусанюк

(подпись) (расшифровка)

Оценка (прописью) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_»\_\_\_\_\_\_\_\_\_\_2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc162626752)

[РАЗДЕЛ 1. ВЕРСТКА САЙТА 5](#_Toc162626753)

[1.1. Этапы разработки сайта 5](#_Toc162626754)

[1.2. Процесс верстки сайта 6](#_Toc162626755)

[РАЗДЕЛ 2. ХОД ВЫПОЛНЕНИЯ ЗАДАНИЯ 8](#_Toc162626756)

[2.1. Разработка HTML разметки 8](#_Toc162626757)

[2.2. Стилизация сайта 10](#_Toc162626758)

[2.3. Разработка логики сайта 12](#_Toc162626759)

[ЗАКЛЮЧЕНИЕ 14](#_Toc162626760)

[Приложение 1 15](#_Toc162626761)

[Приложение 2 16](#_Toc162626762)

[Приложение 3 17](#_Toc162626763)

[Приложение 4 18](#_Toc162626764)

[Приложение 5 19](#_Toc162626765)

[Приложение 6 20](#_Toc162626766)

[Приложение 7 21](#_Toc162626767)

[Приложение 8 22](#_Toc162626765)

[Приложение 9 23](#_Toc162626766)

[Приложение 10 24](#_Toc162626767)

## **ВВЕДЕНИЕ**

Важно отметить, что прохождение производственной практики в студии интернет-решений «Grampus», расположенной по адресу Вологодская область, г. Вологда, ул. Мальцева, д. 52, 2 этаж, офис 208 (Рисунок 1), было для нас очень ценным опытом.



Рисунок 1. Бизнес-центр

Студия, основанная генеральным директором ИП Николаевым Кириллом Александровичем с 2015 года, специализируется на разработке сайтов и оказывает услуги по продвижению компаний в социальных сетях. Их экспертные знания и навыки помогают клиентам получать больше заказов из интернета. Его компания занимается разработкой сайтов, оказанием услуг по продвижению вашей компании в социальных сетях, которые помогут клиентам получать больше заказов из интернета (Рисунок 2).



Рисунок 2. Вход

Продолжительность прохождения практики в студии составляла четыре недели. За это время мы не только закрепили полученную теоретическую базу, но и существенно расширили свой практический опыт работы. Посредством работы над реальными проектами и выполнением реальных задач, основанных на требованиях клиентов, мы научились более глубоко проникать в суть задачи и проявлять креативность в решении проблем. Это позволило нам развить наши навыки в области верстки веб-страниц, анализа и написания кода, а также эффективного использования инструментов для разработки, таких как «Visual Studio Code» и «Figma». Отработка основных принципов работы в этих прикладных средах сделала нас более опытными и уверенными в использовании методов разработки и создания веб-сайтов (Рисунок 3).



Рисунок 3. Офис

До места, в котором мы проходили практику достаточно удобно добраться, вход нее находится на втором этаже. Стол, за которым я работал был вполне удобен, а компьютер достаточно производительный (Рисунок 4).



Рисунок 4. Grampus

Прохождение данной производственной практики ведущей студии интернет-решений «Grampus» позволило нам значительно расширить наши знания и практические навыки в области современной верстки веб-сайтов и разработки. Мы глубже поняли востребованность умений веб-верстки, способность кодировать сайты и эффективно использовать инструменты, так как эти навыки являются неотъемлемой частью современного информационного общества и способствуют нашему более успешному и уверенному участию в практических задачах.

## **РАЗДЕЛ 1. ВЕРСТКА САЙТА**

## **1.1. Этапы разработки сайта**

Верстка сайта, которая включает использование языков разметки HTML и CSS, а также языка программирования для написания интерактивной логики JavaScript, является процессом создания веб-страницы с помощью оформления и размещения контента, чтобы он правильно отображался в браузере.

Важно помнить о дизайне, удобстве использования и доступности для различных устройств и браузеров. Дополнительно, необходимо оптимизировать загрузку страницы для более быстрой отзывчивости и поддержки большего числа устройств. Для этого обычно применяют следующие способы: уменьшение размера изображений, сжатие исходных файлов разметки и кода, кэширование и другие методы.

Верстка является важным этапом разработки веб-сайта, где создается привлекательный и быстро загружаемый контент, доступный для всех пользователей.

**Задание:** Верстка части сайта по дизайну.

Верстка включает в себя следующие этапы:

1. **Создание структуры страницы**: на этом этапе используется язык разметки HTML (Hyper Text Markup Language), который определяет структуру элементов и их иерархию на веб-странице. Заголовки, параграфы, списки, таблицы и другие элементы разметки помещаются в соответствующие теги для определения их семантики и отношений друг с другом.
2. **Оформление и стилизация страницы**: здесь применяется CSS разметка для задания внешнего вида элементов страницы. CSS определяет цвета, шрифты, размеры, отступы, рамки и другие атрибуты стилизации. Стили могут быть заданы для отдельных элементов, классов или идентификаторов, а также использоваться селекторы и правила для группировки стилей.
3. **Адаптивная верстка**: в данном случае, адаптивность означает, что верстка сайта настраивается для оптимального отображения на различных устройствах и экранах. Используется подход «RWD» (Responsive Web Design), где задаются медиа-запросы, которые позволяют странице перестраиваться и адаптироваться к разным размерам экранов. Это позволяет создать удобную и читабельную версию сайта для пользователей мобильных устройств.
4. **Работа с медиа-элементами**: на этом этапе добавляются изображения, видео и аудио на веб-страницу. Для оптимизации загрузки страницы, изображения могут быть оптимизированы по размеру и сжаты. Теги, такие как ***<img>*** для изображений или ***<video>*** для видео, используются для встраивания медиа-элементов на страницу.
5. **Добавление интерактивности**: здесь применяется язык программирования JavaScript для создания интерактивных функций и эффектов на веб-странице. Этот язык позволяет реализовывать анимации, валидацию форм, выпадающие меню, слайдеры и многое другое. С помощью JavaScript можно также обрабатывать действия пользователя и осуществлять взаимодействие с сервером.
6. **Оптимизация загрузки страницы**: верстка также включает в себя методы оптимизации загрузки страницы для улучшения производительности и скорости ее загрузки. Это может включать сжатие CSS и JavaScript файлов, использование кэширования, минимизацию и комбинирование файлов, а также оптимизацию картинок и других ресурсов.
7. **Тестирование и отладка**: на данном этапе осуществляется тестирование верстки на различных браузерах и устройствах для проверки корректности отображения и функциональности. Важно исправить возникающие ошибки и провести отладку, чтобы убедиться, что веб-страница работает должным образом.

## **1.2. Процесс верстки сайта**

1. Ознакомление с макетом сайта в «Figma».

При выполнении работы учитывались требования:

1. Чистая CSS верстка без использования фреймворков («Bootstrap» и прочих);
2. «Hover» эффекты для всех ссылок и активных элементов;

Наш веб-сайт был разделен на шесть основных блоков (Приложение 1), каждый из которых выполняет свою функцию и представляет определенную информацию для посетителей.

1. **Верхняя панель** - верхняя часть нашего веб-сайта, где размещены логотип, навигационное меню и возможно дополнительная информация, такая как контактная информация или ссылки на социальные сети. Верхняя панель помогает пользователям быстро и легко найти нужную информацию и перейти к основным разделам сайта.
2. **Баннер** - представляет визуальное изображение или акцентирует внимание на ключевых элементах сайта. Баннер может содержать важные сообщения, акционные предложения или визуальные элементы, которые привлекают внимание и повышают интерес пользователя.
3. **Товары для дома и дачи по сезонам** - Широкий ассортимент товаров для дома и дачи, подобранный в соответствии с сезонностью. Удобная навигация по категориям для быстрого поиска нужных товаров. Регулярное обновление ассортимента с учетом последних трендов и потребностей покупателей.
4. **Категории товаров** - Качественные фотографии и детальные описания каждого товара. Удобная сортировка товаров по параметрам, таким как цена, материал, цвет и т. д. Возможность сравнения нескольких товаров одновременно для выбора лучшего варианта.
5. **Наши партнеры** - Сотрудничество с надежными поставщиками и производителями, гарантирующими качество товаров. Эксклюзивные предложения и скидки для наших клиентов. Профессиональная помощь и консультации по выбору и приобретению товаров.
6. **Отзывы наших клиентов** - Отзывы реальных покупателей, подтверждающие качество товаров и уровень обслуживания. Возможность оставить отзыв о приобретенных товарах и поделиться своим опытом. Использование отзывов для улучшения качества обслуживания и ассортимента товаров.

Такое разбиение на блоки позволяет организовать информацию на сайте и представить ее в систематическом порядке, что улучшает удобство использования и навигацию для посетителей.

## **РАЗДЕЛ 2. ХОД ВЫПОЛНЕНИЯ ЗАДАНИЯ**

## **2.1. Разработка HTML разметки**

Когда пользователь открывает веб-страницу, его браузер получает HTML код от сервера и анализирует его. Браузер интерпретирует теги HTML и их атрибуты, чтобы определить структуру контента и размеченные элементы страницы. Затем браузер отображает содержимое согласно заданной структуре, применяя стили CSS для придания внешнего вида и поддерживая взаимодействие с помощью JavaScript.

HTML является основным строительным блоком веб-страниц и предоставляет стандартизированный синтаксис разметки. Он обеспечивает доступность, переносимость и совместимость веб-сайтов между разными браузерами и устройствами. HTML играет ключевую роль в создании информационной структуры веб-страниц и формировании пользовательского опыта при просмотре веб-сайтов.

***<head>***: элемент метаданных документа.

HTML-элемент ***<head>*** содержит дополнительную машиночитаемую информацию (metadata) о документе, например его заголовок, скрипты и страницы стилей. Примечание: ***<head>*** в основном содержит информацию для машинной обработки, а не для восприятия человеком.

В данном элементе я подключил:

1. Основной файл с стилями CSS;
2. Файл со стилями векторных иконок из облачного хранилища компании «Google»;
3. Файл со стилями для адаптации сайта на разные устройства;
4. Модуль скрипта для интерактивного слайдера с применением, требуемой в задании, библиотеки «Swiper» (рисунок 5).

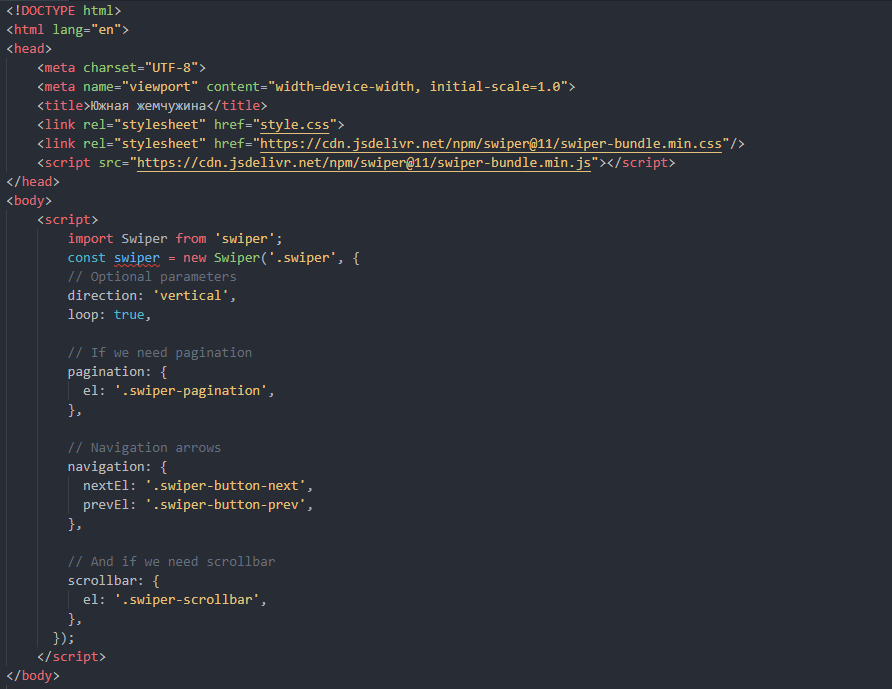


Рисунок 5. Элемент метаданных документа ***<head>***

CSS позволяет мне применять разнообразные стили и эффекты к элементам на странице, улучшая их внешний вид и привлекательность. Благодаря библиотеке «Swiper» я могу создавать интерактивные слайдеры, карусели и другие элементы, обеспечивая удобную навигацию для пользователей и привлекательную представление контента.

***<body>***:элемент, которые представляет собой контент

В стандарте HTML может быть только один элемент ***<body>***, который определяет основное отображаемое содержимое веб-страницы.

Элемент ***<body>*** содержит все отображаемые на странице элементы, такие как контейнеры ***<div>***, абзацы ***<p>***, заголовки ***<h1>*** и ***<h2>***, растровые ***<img>*** и векторные ***<svg>*** изображения, ссылки ***<a>*** и другие элементы, показывая их на экране пользователя в порядке расположения их в разметке или другими различными методами указанными в файле с стилями.

Современные браузеры поддерживают потоковый контент такой как изображения ***<img>***, видео ***<video>*** и другой контент, который нужно загружать. С помощью потокового контента можно создавать веб-страницы с аудио- и/или видеопроигрывателями, которые могут воспроизводить контент до завершения загрузки файла. Это полезно для оптимизации производительности и улучшения пользовательского опыта, поскольку позволяет пользователям начать просмотр контента до того, как весь файл будет загружен.

Открывающий тег элемента ***<body>*** может быть пропущен в случае, если первое, что находится внутри этого элемента, не является пробелом, комментарием, элементом ***<script>*** или элементом ***<style>***. В таком случае, браузер автоматически вставляет открывающий тег ***<body>*** перед этим контентом.

Это правило позволяет упростить написание HTML кода и сделать его более лаконичным. Однако рекомендуется всегда явно указывать открывающий и закрывающий теги для элемента ***<body>*** для более ясного и читаемого кода.

В первом блоке (Приложение 2) в специальном элементе ***<header>*** указываются 4 секции ***<section>***:

1. Лого компании, состоящее из 2 строк;
2. Меню для больших экранов содержащие ссылки на основные блоки страницы;
3. Телефон для связи;
4. Мобильная кнопка открытия меню, которая видна только на маленьком экране и при нажатии на неё показывается элемент с мобильным меню.

Также в первый блок относится кнопка для просмотра квартир в наличии, которая всегда будет в углу экрана.

Во втором блоке (Приложение 3) отображается слайдер, реализованный при помощи библиотеки «Swiper», для этого внутри секции указаны следующие элементы:

1. Контейнер ***<div>*** с атрибутом класса «***swiper-left***», представляет собой кнопку для перелистывания слайдера влево.
2. Контейнер с атрибутом класса «***swiper-wrapper***», нужен для указания библиотеке что в этом контейнере содержатся все слайды. Каждый слайд состоит из двух частей:
   1. Первая часть состоит из логотипа компании, заголовка и кнопки;
   2. Вторая часть состоит из картинки и ссылки снизу картинки.
3. Контейнер практически идентичный первому, только для листания вправо.

В третьем блоке (Приложение 4) содержится кратка информация о доме с 4 фактами о нем. Он поделен на 2 блока:

1. Основная часть, поделенная на картинку дома в правой части и краткая информация о доме в левой части с адресом дома;
2. Снизу находятся блок с фактами.

В четвертом блоке (Приложение 5) показана интерактивная планировка и краткая информация про нее. Поделена также на 2 части:

1. Часть с заголовком, кнопками для переключения между частями планировки и краткая информация про планировку;
2. Изображение планировки, на которой видна только одна сторона, переключаемая при помощи кнопок в 1 части блока.

Пятый блок (Приложение 6) представляет собой информацию о основных преимуществах дома, в нем отображается 8 блоков с заголовком, абзацем и маленькой картинкой в каждом.

В шестом блоке (Приложение 7) показана инфраструктура около дома, содержащая элемента представляющих собой заголовок и картинку

## **2.2. Стилизация сайта**

CSS (Cascading Style Sheets) — это язык стилей, который используется для описания и определения внешнего вида документов (Рисунок 6), написанных с использованием языка разметки, например, HTML. С его помощью можно управлять стилями, цветами, шрифтами, размерами, расположением элементов на странице и другими атрибутами, чтобы создавать эстетически приятные и функциональные веб-страницы.

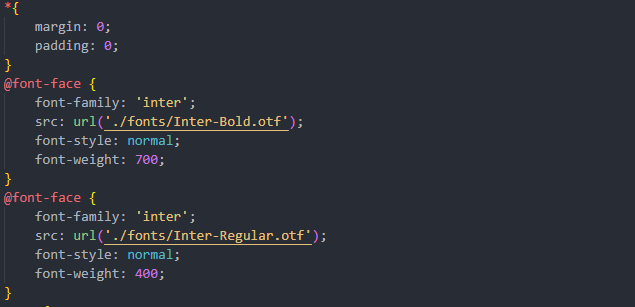


Рисунок 6. Cascading Style Sheets

CSS декларирует стили, которые применяются к конкретным элементам разметки веб-страницы. Декларации стилей состоят из селекторов и объявлений. Селекторы указывают, к каким элементам применяются стили, а объявления содержат инструкции отображения, например, цвет фона, размер шрифта или отступы. Набор правил CSS позволяет гибко задавать стиль и расположение элементов на странице.

CSS также может применяться к другим типам документов XML, таким как SVG (Scalable Vector Graphics) – формат для создания векторной графики, и XUL (XML User Interface Language) – для разработки пользовательского интерфейса. CSS обеспечивает согласованность и отделение оформления от контента в этих документах, позволяя управлять их визуальным представлением.

CSS является важной составной частью веб-разработки и способствует созданию эстетически приятных, функциональных и современных веб-страниц.

Атрибут ***«sticky»*** в CSS используется для создания элементов, которые сначала позиционируются относительно своего обычного положения в потоке, но затем остаются закрепленными на экране при прокрутке страницы. Это особенно полезно для создания навигационных панелей или заголовков, которые остаются видимыми всегда, пока пользователь прокручивает содержимое страницы.

Атрибут ***«position»*** устанавливает позиционирование элемента (Рисунок 7), а ***«sticky»*** указывает, что элемент будет закреплен на экране, если он должен быть виден.

Атрибуты ***«top», «right», «bottom»*** и ***«left»*** использоваться для определения расстояния от соответствующей границы родительского элемента.

Значение ***«sticky»*** не работает в старых версиях Internet Explorer и Edge, но поддерживается большинством современных браузеров, включая Chrome, Firefox, Safari и Opera.

Атрибут ***«font-family»*** в CSS используется для задания шрифта, который будет применяться к элементу или тексту. Значение «Wix Madefor Text» представляет определенный тип шрифта (Рисунок 8), который может использоваться веб-разработчиками, работающими с платформой Wix.

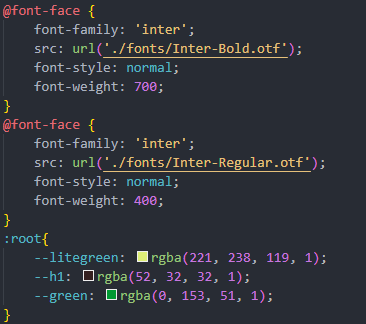


Рисунок 8. Атрибут ***«font-family»***

Атрибут ***«margin»*** в CSS используется для определения отступов элемента от его соседних элементов или от границ родительского элемента. Он может применяться для всех сторон элемента (верх, право, низ, лево) или применяться к каждой стороне индивидуально.

## **2.3. Разработка логики сайта**

Каждый сайт представляет из себя не только текст с картинками, но и интерактивную логику такую как: кнопки, видео, анимации и т.д., которые могут реагировать на действия пользователя, изменять свое состояние в зависимости от разных условий, отображать динамически изменяемую информацию.

Обычно логика сайта создается при помощи языков программирования, раньше так и было, но в современности даже на языке разметке CSS можно создавать простые эффекты такие как: изменение отображения элемента при наведении, зажатии, отпускании мыши, отображение простых видео (слайд-шоу или GIF анимации), простые анимации и другие эффекты, не требующие сложной логики и много зависимостей.

В данном стиле (Рисунок 10) при наведении на любую ссылку (элемент ***<a>***), которая находится в элементе имеющим атрибут с указанным классом «***about-text***», будет изменяться цвет текста данной ссылки. Также в самом элементе, который будет изменяться обычно добавляют стиль «***cursor: pointer;***», который говорит о том, что курсор поменяет стиль.



Рисунок 10. Изменение курсора при наведении мыши

Для создания более сложной логики с применением условий, изменение большого количества элементов или отображения элементов, полученных извне браузера, используется в большинстве случаев скриптовый язык «JavaScript», он не нуждается в компиляции, сложном изучении и код на нем очень компактен. Он применяется в большинстве сайтов, но также есть аналоги от разных компаний такие как «Cmm», «PHP» и т.д., которые не получили такой популярности, хотя PHP все еще довольно часто применяется.

На сайте реализована немного логики, в основном использована сторонняя библиотека «Swiper», которую требуется использовать по заданию, а также некоторые кнопки, функционал которых довольно сложно сделать без JavaScript и не практично на CSS.

## **ЗАКЛЮЧЕНИЕ**

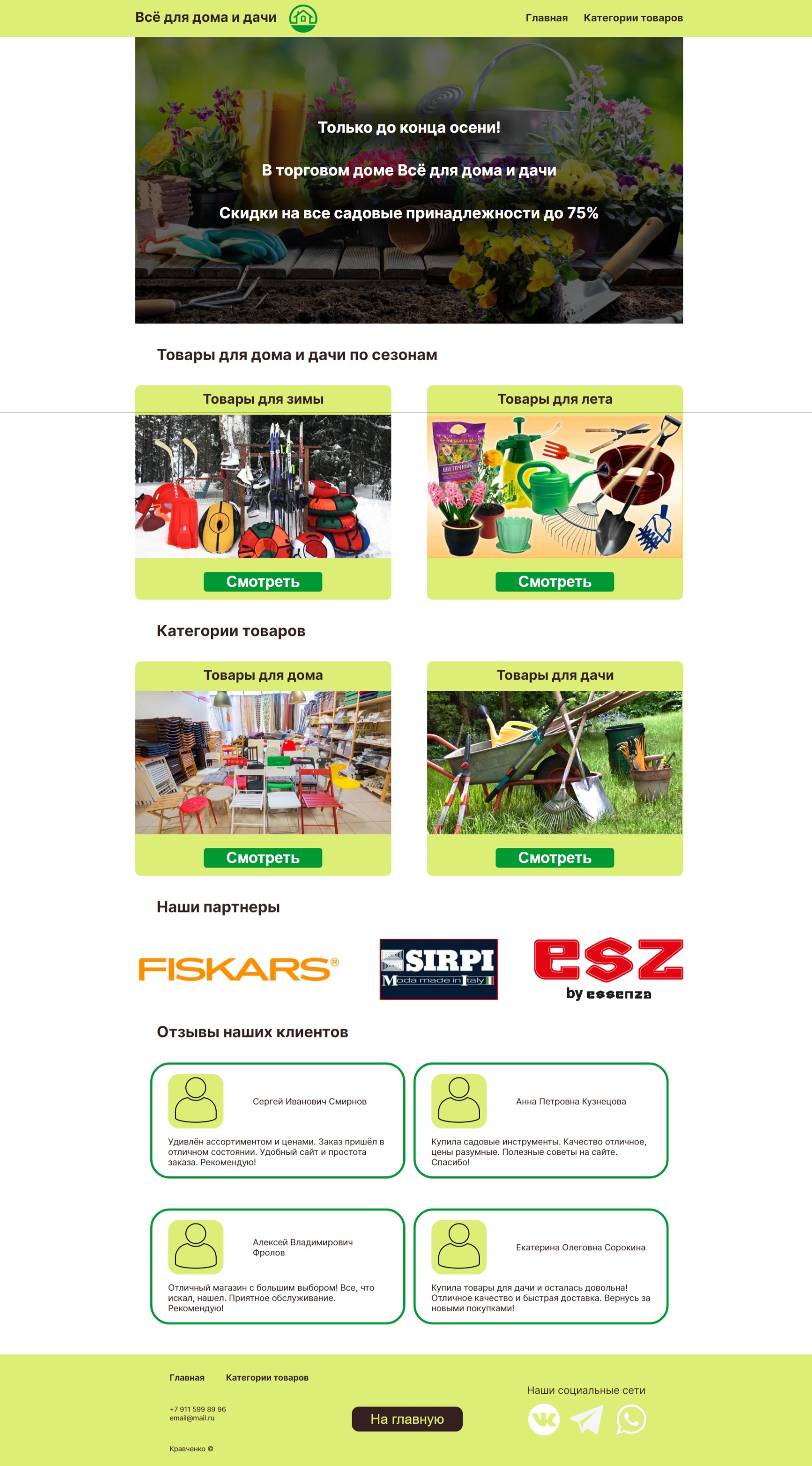
За время производственной практики в студии интернет-решений «Grampus» (г. Вологда, ул. Мальцева, д. 52, 2 этаж, офис 208) я получил ценный практический опыт в области веб-верстки. Я освоил работу с приложениями «Visual Studio Code» и «Figma», что значительно расширило мои навыки и знания, полученные за три года теоретического обучения в колледже. Практика позволила мне закрепить и применить теоретические знания на практике, что оказалось чрезвычайно полезным. Уверен, что полученный опыт поможет мне в дальнейшей карьере.

Я считаю, что подобные практические занятия должны быть неотъемлемой частью учебного процесса, в таком же духе, два раза в рамках каждого курса.

Благодарю команду студии «Grampus» за предоставленную возможность и ценный опыт. Я уверен, что полученные знания и навыки помогут мне достичь успеха в профессиональной деятельности.

# Приложение 1

Дизайн-макет страницы для ПК



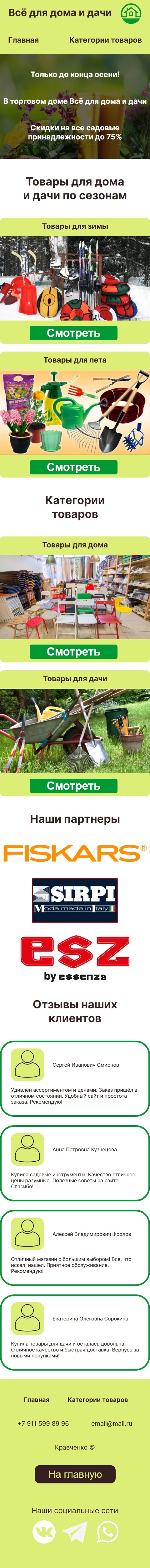
# Приложение 2

Дизайн-макет страницы для планшета



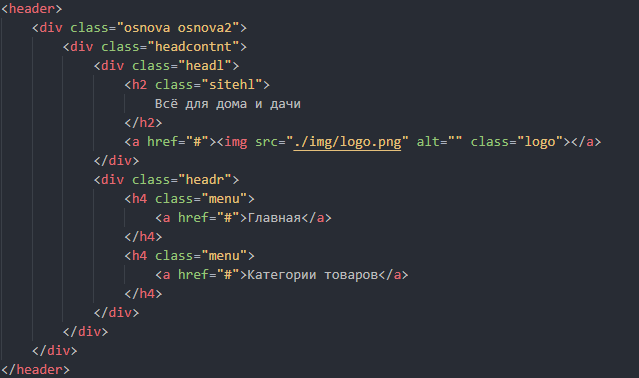
# Приложение 3

Дизайн-макет страницы для телефона



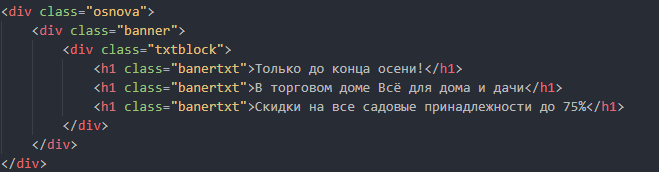
# Приложение 4

Разметка блока «Верхняя панель»



# Приложение 5

Разметка блока «Баннер»



# Приложение 6

Разметка блока «Товары для дома и дачи по сезонам»



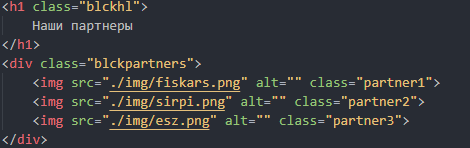
# Приложение 7

Разметка блока «Категории товаров»



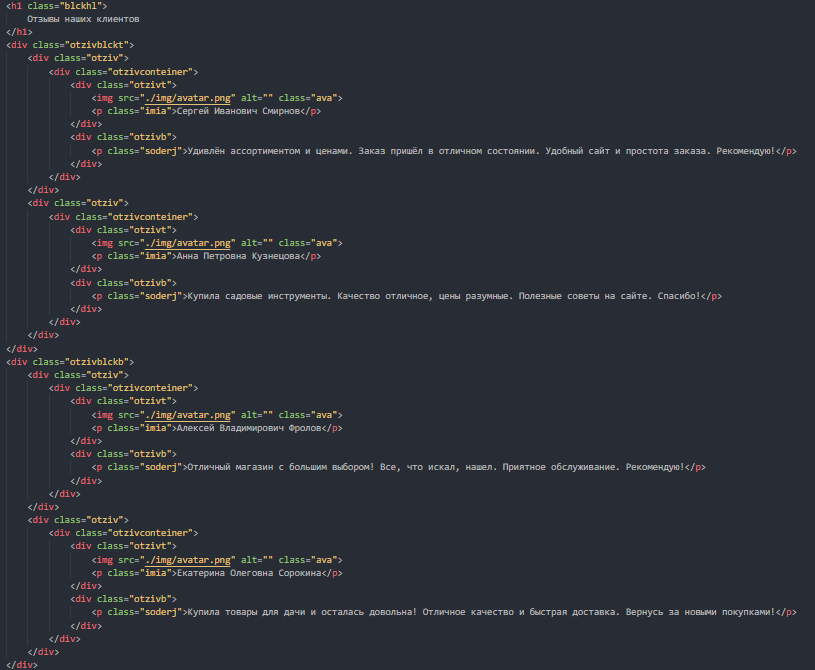
# Приложение 8

Разметка блока «Наши партнеры»



# Приложение 9

Разметка блока «Отзывы наших клиентов»



# Приложение 10

Разметка блока «Подвал»

