МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема: Web-сайт «Сервис для составления графика работы»

**Исполнитель**

студент 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Коренчук

подпись, дата

**Руководитель**

Ассистент каф. ИСИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Харланович

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Харланович

подпись дата инициалы и фамилия

**Содержание**

[Введение 4](#_Toc103468680)

[1 Обзор технических методов и программных средств разработки 5](#_Toc103468681)

[1.1 Обзор аналогичных решений 5](#_Toc103468682)

[1.2 Техническое задание 8](#_Toc103468683)

[1.3 Выбор средств реализации программного продукта 9](#_Toc103468684)

[1.4 Вывод 10](#_Toc103468685)

[2 Макетирование страниц веб-сайта 11](#_Toc103468686)

[2.1 Выбор способа вёрстки 11](#_Toc103468687)

[2.2 Выбор стилевого оформления 11](#_Toc103468688)

[2.3 Выбор шрифтового оформления 12](#_Toc103468689)

[2.4 Разработка логотипа 12](#_Toc103468690)

[2.5 Разработка пользовательских элементов 12](#_Toc103468691)

[2.6 Разработка спецэффектов 13](#_Toc103468692)

[2.7 Вывод 13](#_Toc103468693)

[3 Реализация структуры веб-сайта 14](#_Toc103468694)

[3.1 Структура HTML-документа 14](#_Toc103468695)

[3.2 Добавление таблиц стилей Sass и CSS 17](#_Toc103468700)

[3.3 Использование стандартов XML (SVG) 18](#_Toc103468701)

[3.4 Выводы 19](#_Toc103468702)

[4 Тестирование веб-сайта 20](#_Toc103468703)

[4.1 Адаптивный дизайн веб-сайта 20](#_Toc103468704)

[4.2 Кроссбраузерность веб-сайта 21](#_Toc103468705)

[4.3 Руководство пользователя 23](#_Toc103468706)

[4.4 Выводы 23](#_Toc103468707)

[Заключение 24](#_Toc103468708)

[Список использованных источников 25](#_Toc103468709)

[Приложение 26](#_Toc103468710)

[Приложение A Прототипы веб-страниц 26](#_Toc103468711)

[Приложение Б Макет структуры веб-сайта 27](#_Toc103468712)

[Приложение В Листинг HTML-документа 28](#_Toc103468713)

[Приложение Г Листинг (CSS) 29](#_Toc103468714)

[Приложение Д Листинг XML 32](#_Toc103468715)

[Приложение Е Листинг SVG 33](#_Toc103468716)

[Приложение Ж Листинг JavaScript Slider 34](#_Toc103468717)

[Приложение З Листинг JavaScript Animation 37](#_Toc103468718)

[Приложение К Листинг JavaScript Burger-menu 39](#_Toc103468719)

# Введение

В настоящее время количество новых сайтов увеличивается с неимоверной скоростью. Именно поэтому, основной задачей, ставящейся при разработке сайта – сделать его максимально привлекательным для потенциальных пользователей и придать ему индивидуальность. На заре развития интернета можно было привлечь достаточно большое количество посетителей на сайт, не прикладывая к его разработке больших усилий и без особых материальных затрат. Сейчас же, в условиях возросшей конкуренции, функциональность сайта, его стиль и запоминаемость играют очень большую роль. Ведь независимо от того создается сайт для представления какой-либо фирмы или просто для заработка на рекламе, основной его функцией является привлечение как можно большей аудитории. Поэтому для достижения успеха важен правильный подход к выбору стратегии и исполнителей.

Профессиональные студии веб – дизайна предлагают создание сайтов различного предназначения, это может быть корпоративный сайт, сайт – визитка, интернет – магазин, сайты для ведения бизнеса как онлайн, так и в реальном мире.

Наше время называют «информационным веком». Это название возникло потому, что самым важным, ценным и необходимым ресурсом является информация. Обладание информацией и умение своевременно, быстро и четко донести необходимую информацию до клиентов – ключ к успеху в реалиях современного бизнеса. В настоящий момент именно Интернет, способен оперативно и массово передавать текст, звук, изображения и даже видео-ролики, т.е. практически любую информацию. C точки зрения пользователя, Интернет – это огромный информационный ресурс, в котором можно найти все, что угодно: от прогноза погоды до личных предпочтений голливудских кинозвезд. Аудитория сети Интернет растет с каждым годом. Интернет – самое быстроразвивающееся средство передачи информации за все историю человечества. Пользователи сети интернет являются целью особого интереса для рекламодателей, потому что факт остается фактом – среди пользователей сети куча потенциальных клиентов.

Цель курсовой работы: разработать веб-сайт для онлайн расписаний с использованием HTML5 и XML, а также с применением Sass/CSS3.

Задачи:

* Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
* Разработать макет и прототип сайта;
* Разработать структуру веб-сайта;
* Наполнить сайт информацией по теме;
* Протестировать веб-сайт;
* Разработать руководство пользователя.

# 1 Обзор технических методов и программных средств разработки

## 1.1 Обзор аналогичных решений

Курсовой проект представляет из себя веб-сайт онлайн расписаний. В бизнес-сфере, в частности в сфере онлайн расписаний, уже существуют некоторые аналогичные решения. В этом разделе будут приведены веб-сайты существующих сайтов онлайн расписаний.

Аналог №1 – сайт canva.

Сайт Canva разработан в 2013 году. На рисунке 1.1 представлено оформление сайта. Сайт ориентируется на целевую аудиторию, состоящую из людей, которые хотят структурировать свой распорядок дня, что позволяет минимизировать потери и снизить тревожность.

Шрифт читабельный. Сайт динамический, интерактивный. Также есть анимация при наведении на картинки. Сайт использует три основных цвета (фиолетовый, черный и белый), что положительно сказывается на общем восприятии сайта.

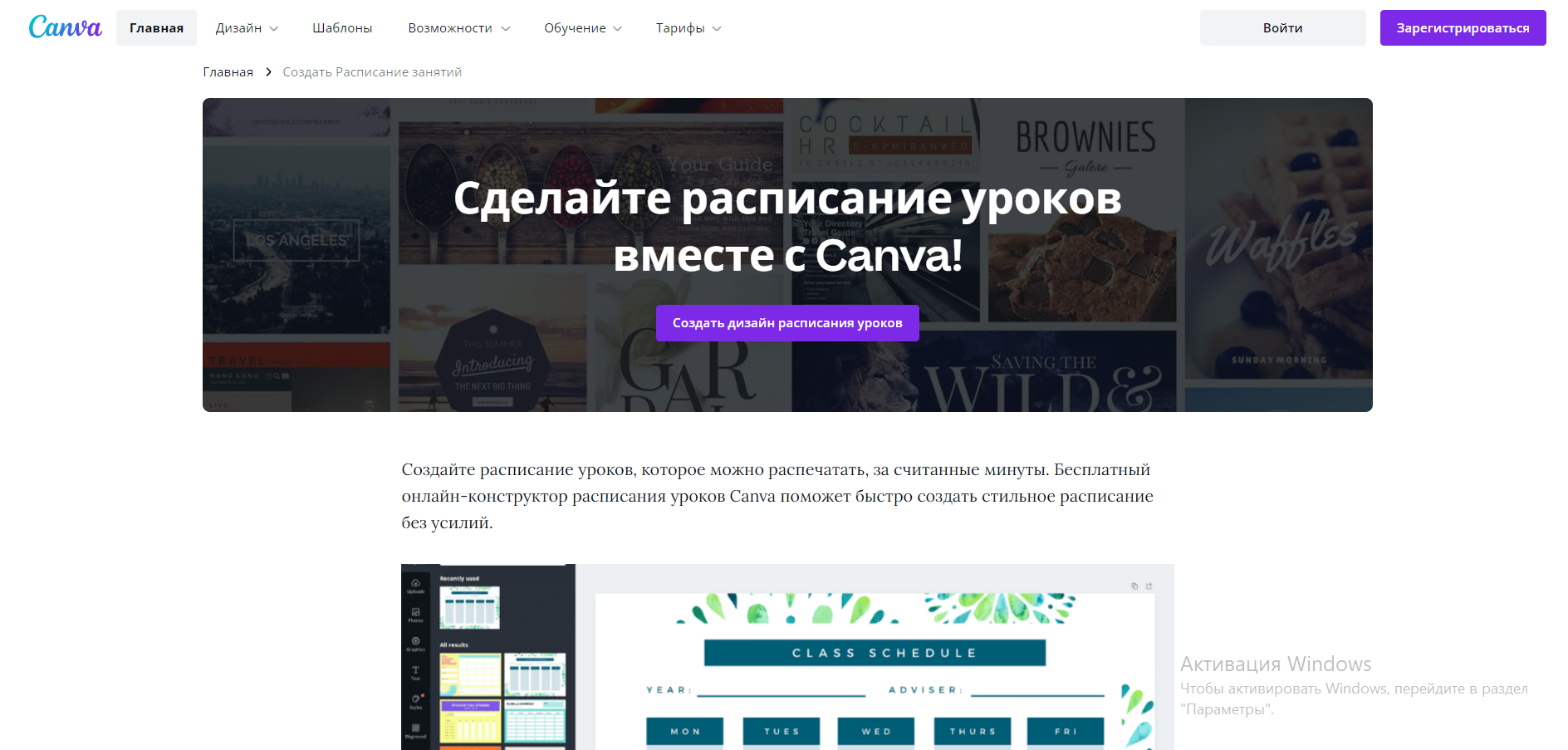


Рисунок 1.1 – Главная страница сайта canva

Также хорошо проработана мобильная версия сайта, горизонтальное меню становится выпадающим, текст остаётся читабельным. Дизайн представлен на рисунке 1.2.

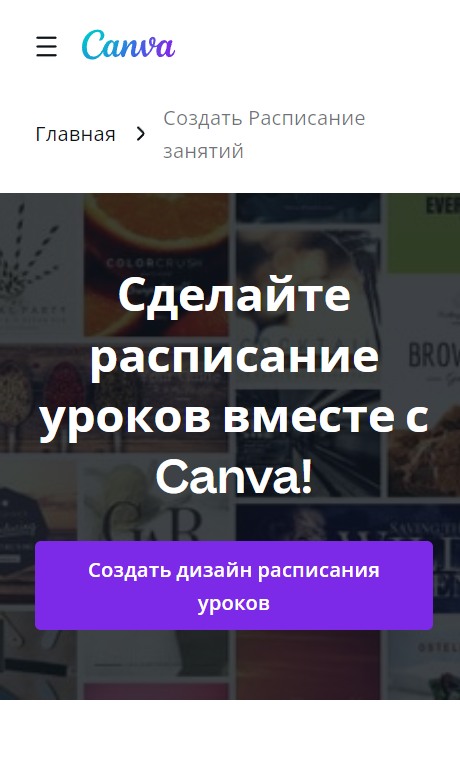


Рисунок 1.2 – Мобильная версия сайта canva

Аналог №2 – сайт vistacreate.

Следующий сайт Vistacreate. Этот сайт имеет приятный вид. Но текст в некоторых местах мелкий. Сайт имеет достаточно информации, но есть немного и лишней. Дизайн представлен на рисунке 1.3.

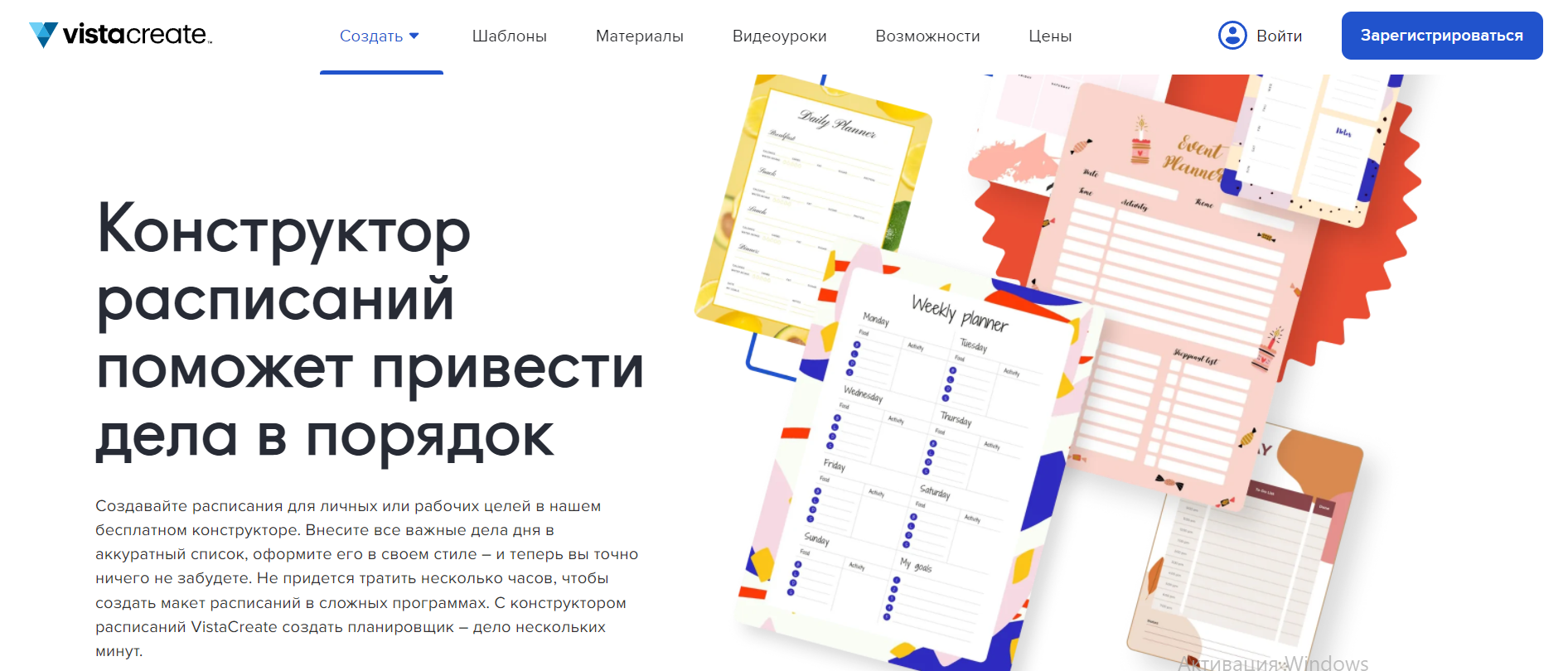


Рисунок 1.3 – Главная страница сайта vistacreate

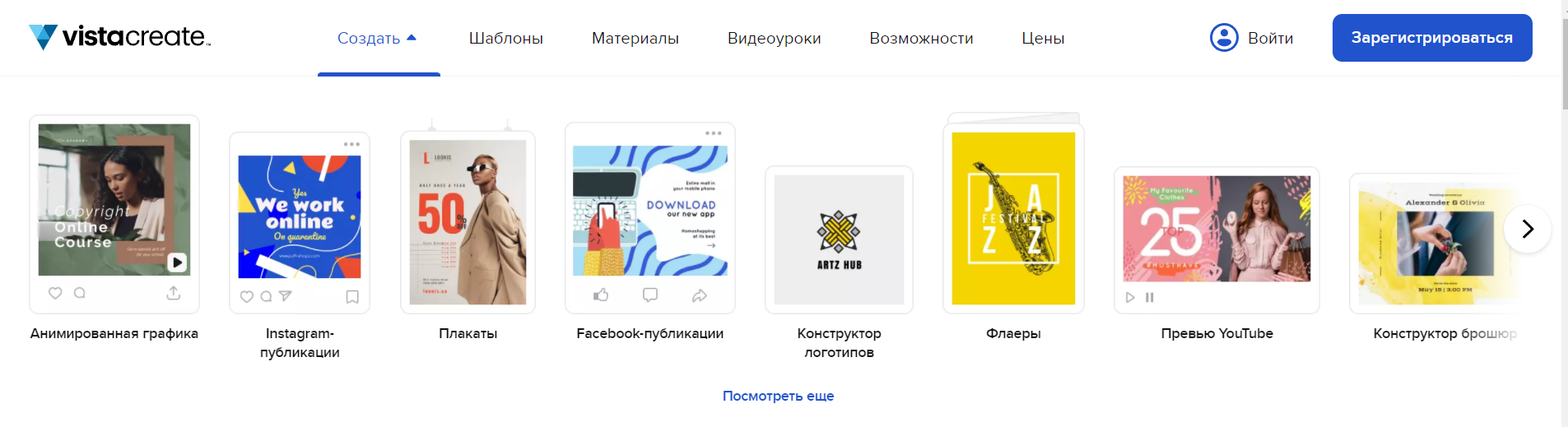


Рисунок 1.4 – Меню главной страницы сайта vistacreate

На рисунке 1.4 видно, что при наведении на кнопку навигационного меню данная кнопка начинает подсвечиваться. Так как меню выпадающее, при нажатии на кнопку появляется информация и слайдер. При анализе данного ресурса складывается впечатление, что все элементы сайта не структурированы и размещены хаотично. Данный сайт обладает кроссбраузерностью и адаптивной версткой.



Рисунок 1.5 – Мобильная версия сайта vistacreate

У данного сайта нет никаких проблем с адаптивностью, на разных устройствах сайт выглядит привлекательно. На рисунке 1.5 представлено оформление на мобильных устройствах. Видно то, что интерфейс немного отличается. В связи с нехваткой места на мобильных устройствах меню перенесли в левый верхний угол. В целом, сайт на мобильном устройстве представлен также, как и на персональном компьютере.

Также, с помощью расширение которое есть в браузере «Google Chrome», я выяснил: помимо HTML5, CSS и JavaScript, использует ещё и дополнительные фреймворки или библиотеки, которые представлены на рисунке 1.6. По подробнее о каждом инструменте создания сайтов:

Так, например, на сайте «Vistacreate» присутствует фреймворк jQuery.

JQuery-набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

AJAX –подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. По-русски иногда произносится транслитом как «аякс».

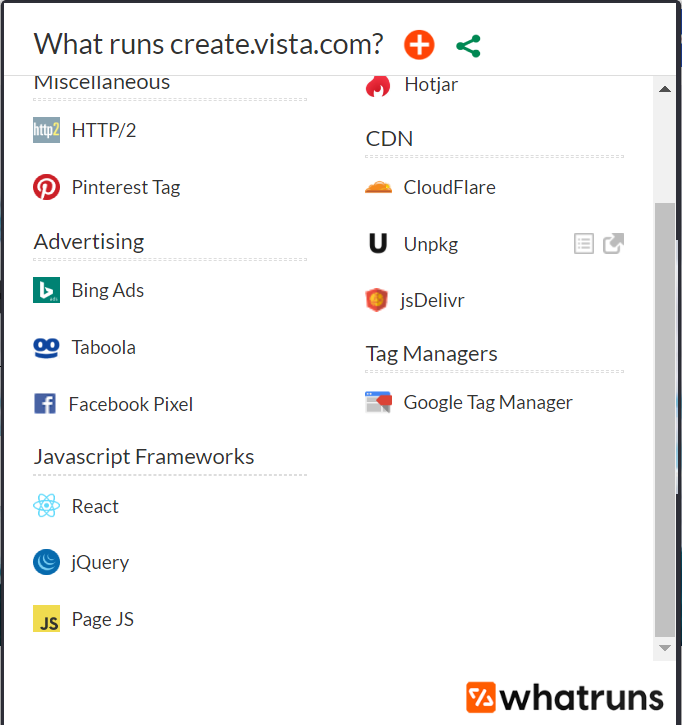


Рисунок 1.6 – Окно с используемых технологиями сайта createvistra

React — это JavaScript-библиотека для создания пользовательских интерфейсов. Обратите внимание, что это именно библиотека, а не фреймворк. React часто называют фреймворком, но это ошибка. Во-первых, его использование ни к чему вас не обязывает, не формирует «фрейм» проекта. Во-вторых, React выполняет единственную задачу: показывает на странице компонент интерфейса, синхронизируя его с данными приложения, и только этой библиотеки в общем случае недостаточно для того, чтобы полностью реализовать проект.

Аналог №3 – сайт planimum.



Рисунок 1.6 – Главная страница сайта planimum

На рисунке 1.6 представлено оформление сайта planimum.ru. Текста очень мало. И дизайн оставляет желать лучшего. Очень мало текста, мало динамически эффектов. Цвета не сочетаются друг с другом.

## 1.2 Техническое задание

В данном проекте требуется создать одностраничный сайт онлайн расписаний. Сайт должен содержать общую информацию онлайн расписаний, такую как контактные данные, описание компании, а также возможности онлайн расписаний. Основным контентом являются текстовые данные, фотографии.

Главной задачей сайта является предоставление данных о возможности онлайн расписаний. Сайт должен быть интерактивным, содержать актуальную информацию. Данный сайт предназначен для желающих контролировать свое расписание.

В мобильной версии (ширина до 560px) страница должна адаптироваться под ширину устройства. Также должна быть разработана версия для планшетного устройства (ширина до 1000px). Начиная с ширины в 1000px должна открываться версия для персонального компьютера.

На странице «Главная» должна быть расположена основная информация, акционные баннеры.

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа, адаптивность и кроссбраузерность сайта (корректное отображение в браузерах, таких как Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.). В проекте используется гипертекстовый язык разметки HTML, CSS/Sass, графические элементы в формате SVG; данные хранятся в XML-формате.

Задачи курсового проекта:

- Размещение основной информации на сайте.

- Размещение информации о товаре.

- Представление имиджа компании.

## 1.3 Выбор средств реализации программного продукта

Сайт разрабатывался в текстовом редакторе Visual Studio Code. Страницы сайта могут просматриваться в таких браузерах как Internet Explorer, Microsoft Edge, Safari, Opera, Mozilla Firefox, Google Chrome.

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS, представлен на рисунке. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга.

Для реализации проекта были выбраны языки: HTML, CSS/Sass, XML, JavaScript. Структура сайта создана с помощью языка разметки HTML, дизайн сайта оформлен с помощью CSS/Sass. В XML формате хранятся данные сайта. JavaScript позволит добавить динамичности сайта, путем введения функций и событий. Подробнее про каждый из них:

**HTML** (от английского**HyperText Markup Language**) — это код, который используется для структурирования и отображения веб-страницы и её контента. Например, контент может быть структурирован внутри множества параграфов, маркированных списков или с использованием изображений и таблиц данных. **HTML** используется, чтобы сообщать вашему браузеру, как отображать веб-страницы, которые вы посещаете. Браузер интерпретирует HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т.д. В 2014 году был создан HTML5. Разработчики нового языка ориентировались на устранение проблем предыдущих версий и реализацию современных функций:

* Высокая скорость работы сайта за счет удаления всего лишнего из кода и перераспределения функциональных элементов.
* Сайты на HTML5 используют совместно ресурсы браузеров и удаленных серверов, чтоб делает серфинг и просмотр сайтов более комфортным.
* Сайты стали более легкими, что очень важно для мобильного интернета, который работает медленнее стационарного.
* Медиа и интерактивные функции сайтов теперь решаются не путем установки на компьютер специальных приложений, а ресурсами самого кода сайта.

Улучшены возможности по работе сайтов на разных операционных системах и с разными браузерами.

CSS (Каскадные таблицы стилей) позволяет создавать великолепно выглядящие веб-страницы, но как же это работает? Эта статья объясняет, что такое CSS, с помощью простого примера синтаксиса, а также охватывает некоторые ключевые термины о языке.

Sass – это своего рода расширение, созданное для упрощения каскадных таблиц стилей (CSS). Все те, кто начинает заниматься разработкой и администрированием сайтов.

JavaScript – это язык программирования, который добавляет интерактивность на ваш веб-сайт (например: игры, отклик при нажатии кнопок или при вводе данных в формы, динамические стили, анимация). Эта статья поможет вам начать работать с этим захватывающим языком и даст вам представление о том, на что он способен.

XML - это язык разметки подобный HTML. Расшифровывается как (англ. Extensible Markup Language - Расширяемый Язык Разметки).

## 1.4 Вывод

В данном разделе были рассмотрены аналогичные решения, были проанализированы их недостатки и преимущества, для создания в последующем качественного веб-сайта. Были поставлены конкретные задачи для создания интерактивного информационного ресурса. Кроме того, были рассмотрены средства реализации программного продукта, такие как языки разметки, а также редактор кода VS Code. Были проанализированы особенности языков и основные дополнительные средства редактора. Все вышеперечисленные пункты понадобятся для достижения поставленной задачи и для создания в дальнейшем качественного продукт.

# 2 Макетирование страниц веб-сайта

## 2.1 Выбор способа вёрстки

В соответствие с требованиями к проекту, в частности адаптивность и кроссбраузерность сайта, было принято решение об использовании SASS.

Для создания отдельных блоков страницы, такие как горизонтальное меню, будет использована Flexbox-верстка. С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров. CSS Flexbox поддерживается всеми используемые на сегодняшний момент современными браузерами (с использованием префиксов: IE10+, Edge12+, Firefox 2+, Chrome 4+, Safari 3.1+, Opera 12.1+, iOS Safari 3.2, Opera mini, Android 2.1+, Blackberry 7+).

Исходя из выбора способа верстки были расположены все основные элементы сайта и разработаны макеты страниц. Макеты смотреть в [приложении Б](#_Приложение_Б_Макет).

Первым шагом было принято создать прототип сайта. По прототипу будет понятно, где будут располагаться основные блоки, картинки, текст. Прототип будет представлен в [приложении А.](#_Приложение_A_Прототипы)

Вторым шагом было принято решение создать основной макет. Макет будет состоять из основных блоков, таких как header, main, section, footer. Макет будет представлен в [приложении Б](#_Приложение_Б_Макет).

Блок header будет фиксированным и при скролле он будет в области видимости. В блоке main будет основная информация и также блок с кнопкой и тексом, а на заднем фоне будет картинка. В блоке section будет секция, в которой будет обязательно блок с текстом и фотографией. А в блоке footer будет пункты меню и авторские права.

## 2.2 Выбор стилевого оформления

Для реализации стилевого оформления сайта было выбрано минималистическое направление с элементами классического веб-дизайна. Такой стиль позволяет привлечь внимание новых пользователей.

Сайт не загромождается лишними деталями (боковыми панелями и т.п.), большим количеством спецэффектов. Акцент на сайте ориентируется на основной контент.

Сайт не загромождается лишними деталями и большим количеством спецэффектов.

Большинство элементов выдержаны в одном стиле.

Цветовые решения не яркие, довольно нейтральные. Основные цвета: #29AAE3, #000000, # 9033F9. Цветовая палитра представлена на рисунке 2.1

|  |  |  |
| --- | --- | --- |
|  |  |  |

Рисунок 2.1 – Картинки с основными цветами на сайте

Эти цвета были подобраны так, потому что они будут приятны для глаз и это цвета, на которое пользователь сразу обратит внимания на сайт.

## 2.3 Выбор шрифтового оформления

В данном проекте будут использован один основной шрифт: Roboto. Шрифт подключался с помощью библиотеки Google Fonts.

Шрифт Roboto является вариативным шрифтом и относится к категории гротески. Гарнитура содержит 2 файла. Поддерживает 71 язык. Относится к семейству шрифтов Roboto. Имеет лицензию Apache 2.0. Можно использовать в коммерческой и не коммерческой деятельности. Разработка шрифта Roboto велась Christian Robertson в студии типографики Google.

## 2.4 Разработка логотипа

Разработка логотипа осуществлялась в графическом редакторе Figma. Дизайн довольно минималистичен; было принято решение не загромождать его множеством декоративных элементов. Логотип представляет из себя название

Shifton представлена на рисунке 2.2.



Рисунок 2.2 – Логотип сайта shifton

Логотип имеет иконку календаря и букв, которые будет принимать определенный шрифт и цвет.

## 2.5 Разработка пользовательских элементов

На сайте присутствуют анимации для того, чтобы сделать сайт более «живым» и привлекательным.

В проекте будут представлены такие элементы пользовательского интерфейса, как кнопки, scroll, слайдер.

Слайдер будет на рисунке 2.3.

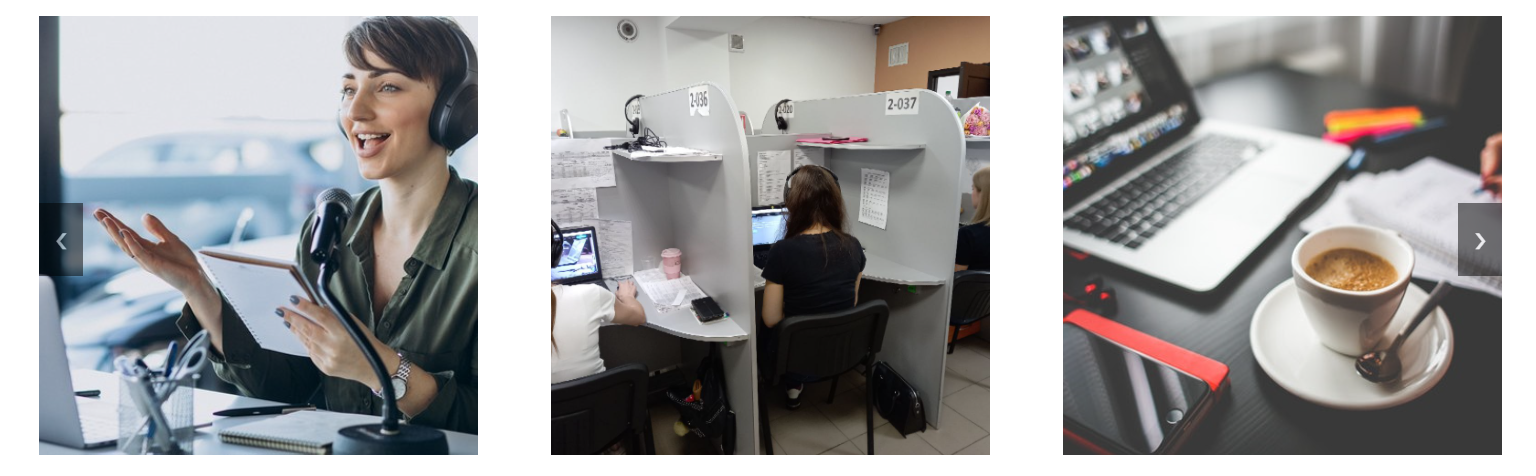


Рисунок 2.3 – Адаптивный слайдер для сайта

Слайдер будет адаптивным под разные устройства. Также он будет иметь плавный переход на следующий слайд. Слайды можно будет перелистывать как и при нажатии на кнопки, так и перелистыванием.

## 2.6 Разработка спецэффектов

На сайте будут разработаны некоторые динамические эффекты и анимации, которые позволят придать сайту интерактивность и сделают его более привлекательным для пользователя.

На главной странице будет присутствовать элемент scroll представлена на рисунке 2.4.



Рисунок 2.4 – Кнопка scroll

Scroll- это способ перемещения визуальной части окна вверх, вниз, вправо и влево для того чтобы увидеть дополнительную информацию в окне.

Кнопка scroll будет на всех устройствах и при нажатии на эту кнопку, скролл будет подниматься в вверх.

## 2.7 Вывод

На данном этапе были созданы прототипы, а затем и макеты сайта для облегчения последующей верстки. Было упорядочено расположение всех элементов на страницах (навигационного меню, логотипа, основного контента каждой страницы, расположение картинок, подвала сайта). Было определено стилевое оформление сайта, то есть основная цветовая гамма, шрифтовое оформление, элементы пользовательского интерфейса, а также эффекты и переходы, придающие динамики страницам.

# 3 Реализация структуры веб-сайта

## 3.1 Структура HTML-документа

Структура HTML-документа определяет базовый «скелет» для будущего веб-сайта. Код выполнен по стандарту HTML5.

Структура тела документа состоит из семантических тегов header, section, article main, footer и блочных тегов div. Листинг представлен в [приложении В](#ПР_3).

В теге header находится горизонтальная навигационная панель. Она содержит ссылки на все основные разделы сайт, логотип и название компании. Также в теге header находиться тег span. Который появляется при уменьшении размера страницы. Элементы навигационного меню автоматически скрываются, а на их месте появляется так называемое “бургер-меню”. Навигационное меню было реализовано при помощи маркированного списка. Код тега header показан в листинге 3.1.

|  |
| --- |
| <!-- header -->     <header class="header">        <div class="container">           <div class="wrapper">              <div class="header\_blocks">                 <div class="logo">                    <a> <img style="min-width: 100px" src="Pictures/logo.svg"></a>                 </div>                 <div class="header\_menu">                    <ul class="menu">                       <li><a class="close\_slider">Главная </a></li>                       <li><a class="open\_blog">Блог</a></li>                       <li><a class="sign-btn">Вход</a></li>                    </ul>                    <div class="regi">                       <a class="menu\_registration">Регистрация</a>                       <div class="hide">                          <div class="container" id="click">                             <div class="bar1"></div>                             <div class="bar2"></div>                             <div class="bar3"></div>                          </div>                       </div>                    </div>                 </div>              </div>           </div>        </div>     </header> |

Листинг 3.1 – Структура «Хедера» web-страницы

В теге section находиться карточки с возможностями, в которых есть кнопка, по нажатию будет выводиться подробная информация. Также этот тег будет содержать текст и картинку. В блоке с текстом будет кнопка, по которой можно будет перейти на слайдер. На листинге 3.2 продемонстрирован код тега section.

|  |
| --- |
| <section class="section\_opportunities">           <div class="container">              <div class="wrapper">                 <h2 class="opportunities">                    Возможности Shifton                 </h2>                 <div class="blocks\_opportunities"…</div>    </div>           </div>        </section> |

Листинг 3.2 – Структура тега section

В теге main находиться информация описание сайта. Там будет на заднем фоне картинка, а в содержании текст и кнопка, при нажатии на которую будет переходить на слайдер. На листинге 3.3 продемонстрирован код тега main.

|  |
| --- |
| <!-- main -->        <main class="main">           <div class="container">              <div class="wrapper">                 <div class="main\_\_osnova\_\_info">                    <h1 class="info">                       Автоматизированное составление<br> рабочих расписаний онлайн                    </h1>                    <p>Экономия времени и денег</p>                    <p>Грамотное распределение рабочей нагрузки                    </p>                    <p>Построение оптимального сменного рабочего<br> расписания для любого количества сотрудников</p>                    <a href="#" class="btn open\_slider">Попробовать бесплатно</a>                 </div>              </div>           </div>        </main> |

Листинг 3.3 – Структура тега main

В теге footer находятся дополнительные ссылки на различную информацию. На листинге 3.4 показан код тега footer.

|  |
| --- |
| <!-- footer -->     <footer>        <div class="container">           <div class="wrapper">              <ul class="nav\_footer">                 <li> <a class=" open\_slider">Возможности</a></li>                 <li> <a class="open\_blog">Блог</a></li>                 <li> <a href="#">Главная</a></li>              </ul>              <hr>              <div class="google\_apple">                 <a href="https://play.google.com/store?hl=ru&gl=US"><img src="Pictures/icons\_social/google\_play.svg"                       alt="google"></a>                 <a href="https://www.apple.com/ru/app-store/"><img src="Pictures/icons\_social/App\_Store.svg"                       alt="App\_Store"></a>              </div>              <h3>Copyright 2017-2021, Shifton. All rights reserved.<br>                 Privacy Policy | Terms of Use | Сookies              </h3>           </div>        </div>     </footer> |

Листинг 3.4 – Структура footer web-страницы

Для реализации блока возможностей было использовано пункты меню сделать через список,а чтобы сделать разные цвета списка и текста был использован тег span. На листинге 3.5 показан код блока возможностей.

|  |
| --- |
| <div class="block\_opp">                       <div class="blocks\_img">                          <img src="Pictures/icons/iconFirst.svg" alt="Картинка" class="img">                       </div>                       <h3>Распределяй <br>работу</h3>                       <p class="plans">Планирование смен                       </p>                       <ul class="information\_about\_blocks">                          <li><span>Настраиваемые шаблоны смен</span></li>                          <li><span>Обмен сменами между сотрудниками</span></li>                          <li><span>Праздничные и особые дни</span></li>                          <li><span>Свободные смены</span></li>                          <li><span>Соблюдение трудового законодательства</span></li>                          <li><span>Синхронизация с календарем</span></li>                       </ul>                       <div class="button\_more">                          <a class="btn\_detailed open\_app">Подробнее</a>                       </div>                    </div> |

Листинг 3.5 – Структура блока кода возможностей

Для реализации слайдера было принято решение о создание слайда, при наведение на который нам будем показывать определённый текс. На листинге 3.6 показан код блока скролла.

|  |
| --- |
| <div class="slide">                    <div class="hide\_slide\_text">                       <h1>Почему Shifton - лучшее решение для любого колл-центра</h1>                       <p>При выборе программы для колл-центра важно учитывать 7 факторов, которые гарантируют                          прибыльность бизнеса. Сегодня мы расскажем, почему в 2021 году Shifton - лучший сервис для                          аутсорсинговых компаний.</p>                    </div>                    <img class="img" src="Pictures/slider/slidetwo.jpg" alt="slide">                 </div> |

Листинг 3.6 – Структура блока кода скролла

Для анимации блоков при скролле было принято решение писать на native js. Это реализовано было так, чтобы первоначально скрывать блоки, которые анимировать. А затем при скролле, когда блоки становиться в области видимости, то показывать их при плавной анимации. Можно посмотреть в [приложении З](#_Приложение_З_Листинг).

Для меню-бургера, который будет работать на мобильных устройствах, было принято решение при нажатии на бургер, будет выдвигаться меню, в которой будут основные пункты меню. Можно посмотреть в [приложении К](#_Приложение_К_Листинг).

Для слайдера было принято решение о переключение слайдов не только по кнопкам, но и перелистывании. Слайдер будет иметь 3 слайда на больших экранах, на средних экранах 2, а на мобильных устройствах будет 1 слайд. Можно посмотреть в [приложении Ж.](#_Приложение_Ж_Листинг)

## 3.2 Добавление таблиц стилей Sass и CSS

Добавление таблиц стилей Sass(Scss) и Css позволило упростить создание стилистического оформления страницы. Компиляция осуществлялась благодаря open-source приложению Preprose. Преимуществами данного приложения являются компиляция файлов sass, less, scss и т.д. Приложение не зависит от того, установлены ли в системе соответствующие программы и библиотеки (sass, compass и т.п). Обновление страниц в браузере в реальном времени и конечно http-server. При активированной опции автокомпиляции после каждого сохранения файла программа выводит информацию о проведенной обработке файлов.

Главными преимуществами Sass(Scss) перед Css это прежде всего автоматическая кроссбраузерность при компиляции Sass. Наличие вложенности в отличие от Css. Например, Sass облегчает применение свойств благодаря миксинам. Они позволяют один раз создать набор правил, чтобы потом использовать их многократно или смешивать с другими правилами. Переменные, управлять которыми намного проще в отличие от Css. А также импорт, Sass делает это без HTTP-запросов. Листинг представлен в [приложении Г](#_Приложение_Г_Листинг).

В данной структуре используется внешнее подключение таблиц стилей через тег link, так как это самый мощный и удобный способ определения стилей и правил для веб-сайта. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. В таблицах стилей используются селекторы по классу, по идентификатору, универсальный селектор, а также селекторы потомков, псевдоклассы и псевдоэлементы.

Объявление стиля состоит из двух частей: селектора и объявления. Объявление состоит из двух частей: имя свойства (например, color) и значение свойства (grey). Селектор сообщает браузеру, какой именно элемент форматировать, а в блоке объявления (код в фигурных скобках) перечисляются форматирующие команды — свойства и их значения.

## 3.3 Использование стандартов XML (SVG)

Было принято решение о работе с XML форматом. Главной причиной этому является простота использование данных, но легче в использовании был бы Json формат. К преимуществам формата Json можно отнести-удобочитаемость кода, простота создания объекта данных на стороне сервера и обработки данных на стороне клиента, возможность расширения, отладки и исправление ошибок, а также безопасность.

В проекте формат XML был использован для хранение данных о постах. Листинг предоставлен в [приложении Д](#_Приложение_Д_Листинг).

XML документ использовался для описания всех предложенных курсов. В нем использовались следующие теги:

* <root> (тег-контейнер, содержащий все теги и весь XML файл);
* <string> (тег-контейнер, содержащий другие теги, представленные ниже, в которых буду содержатьcя в блоге) ‘
* <id> (содержащий id блога);
* <title> (тег, содержащий заголовок блога);
* <description> (тег, содержащий описание блога);
* <img> (тег, содержащий картинку блога);

Для внедрения иконок было принято решения использовать расширение svg. Что позволило изменять размер экрана, при этом не изменить качество изображения. Листинг предоставлен в [приложении Е](#_Приложение_Е_Листинг).

Формат js был использован для создания анимации во время определенных событий. В навигационном меню присутствует меню-бургер. Нажав на него срабатывает скрипn приведенный в [приложении Ж](#_Приложение_Ж_Листинг). Также формат помог реализовать анимацию при скролле и слайдер. Скрипты приведенные в [приложении Ж](#_Приложение_Ж_Листинг).

## 3.4 Выводы

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей CSS/Sass. Было продемонстрировано использование стандартов JSON (SVG), а также языка программирования JS. Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

# 4 Тестирование веб-сайта

## 4.1 Адаптивный дизайн веб-сайта

Сегодня количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. А потому адаптация веб-сайта под маленькие экраны – актуальная задача и для курсового проекта.

Передо мной стояла задача: создать сайт, который будет сам подстраиваться под разные типы экранов. Главными условиями адаптивности является не фиксированный размер элементов страницы.

Для тонкой настройки элементов во время уменьшения экрана были использованы медиа запросы. Листинг медиа запросов представлен в [приложении Г](#_Приложение_Г_Листинг).

После написания основной структуры страницы на HTML и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров. После этого было выявлено, что во всех браузерах сайт выглядит одинаково.

При отображении сайта на мобильных устройствах навигационное меню в верхнем колонтитуле меняется на выпадающее меню, открывающееся при нажатии на кнопку меню. На рисунке 4.1 можно увидеть пример адаптивного дизайна для мобильных устройств.

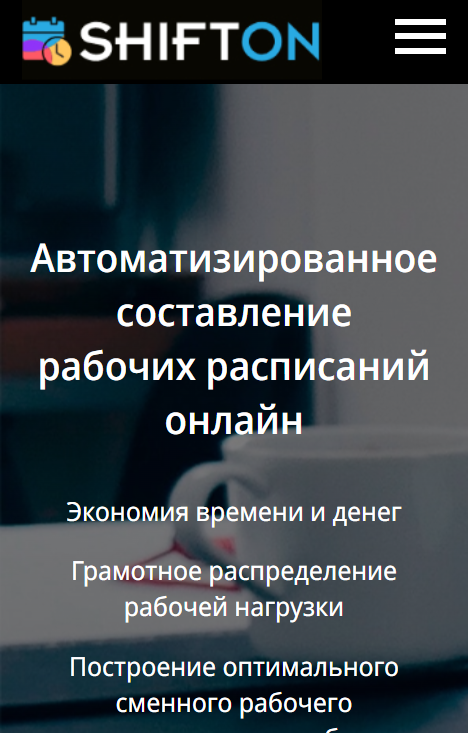


Рисунок 4.1 – Адаптивный главной страницы для телефонов

Также при использовании мобильного устройства блоки с возможностями на странице для отображения располагаются друг под другом. На рисунке 4.2 можно увидеть пример.



Рисунок 4.2 – Блоки с возможностями на мобильных устройствах

И таким образом можно сказать, что все блоки располагаются друг за друг, потому что пользователю было все хорошо видно на его мобильном устройстве.

## 4.2 Кроссбраузерность веб-сайта

Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали было принято решения о внедрении вендорных префиксов. Они были автоматически прописаны после компиляции файла с расширением sass.

Веб-сайт будет тестироваться в следующих браузерах: Google Chrome, Fire Fox, Opera и Microsoft Edge.

При проверке веб-сайта в Google Chrome никаких искажений текста, анимации, переходов не было выявлено. На рисунке 4.3 можно увидеть пример веб-страницы в Google Chrome.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рисунок 4.3 – Вид веб-страницы в Google Chrome а) мобильная версия; б) компьютерная версия

В Fire Fox сайт показал схожий результат: никаких отклонений как для мобильной версии сайта, так и для компьютерной не было выявлено, текст не съехал, не произошло его наложения на картинки. На рисунке 4.4 можно увидеть пример веб-страницы в Fire Fox.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рис 4.4 – Вид веб-страницы в Fire Fox а) мобильная версия; б) компьютерная версия

В Opera всё так же работает без неисправностей, весь контент обладает правильным шрифтом, не наблюдается сдвиг текста или отсутствие картинок. На рисунке 4.5 можно увидеть пример веб-страницы в Opera.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рис 4.5 – Вид веб-страницы в Opera а) мобильная версия; б) компьютерная версия

При просмотре веб-сайта с помощью таких браузеров как Google Chrome, Microsoft Edge и Mozilla Firefox общий вид веб-страницы не изменился.

## 4.3 Руководство пользователя

Данный веб-сайт предоставляет информацию пользователю о сервисе для составления графика работы и позволяет сделать свой график более логичным.

Для обеспечения надежности данного веб-сайта для компании “Shifton” реализовано:

- разработан удобный пользовательский интерфейс, чтобы даже человек, не работавший с компьютером достаточно, быстро мог разобраться в этом веб-сайте;

Вверху главной страницы, вы можете увидеть навигационную панель. Она содержит основные тематические разделы веб-сайта: “Главная”, “Блог”.

Во вкладке “Главная” пользователь может изучить всю основную информацию об этом сервисе.

Во вкладке “Блог” пользователь может получить краткую информация о изменениях в календаре.

## 4.4 Выводы

Цель тестирования Главной целью такой тщательной проверки сайта является грамотная настройка всех показателей, одной или всех страниц сайта, оперативное обнаружение и устранение всех дефектных мест на сайте, а также его адаптация под различные устройства.

Динамика сайта на разных устройствах – наиболее важная часть в написании сайта, поэтому, на данном этапе было проанализировано поведение веб-сайта при использовании с разных устройств, а также веб-ресурс был протестирован в большом количестве браузеров. По итогу выполнения данного этапа для веб-сайта был разработан адаптивный дизайн и кроссбраузерность. Работа, проделанная с сайтом, позволяет пользователям ориентироваться по сайту. Лишних элементов нет, что упрощает пользование сайтом. При пользовании сайтом при помощи мобильного устройства некоторые элементы меняют свое положение, для более удобного пользования. В ходе тестирования были проверены все ссылки (включая модальные окна), проверены формы регистрации пользователя. Была выполнена проверка контента. Также в ходе тестирования была проверена совместимость с браузерами. (Google Chrome, Opera, Firefox, Microsoft Edge). В браузере Internet Explorer тестирование давала некоторые сбои. А именно проблема с запуском с открытого сервера (live server), а также анимации сделанные непосредственно через JavaScript ввели себя некорректно.

# Заключение

Целью на курсовое проектирование было создание сервиса для онлайн расписаний «Shifton». Создание веб-сайта, который в основном облегчит сделать свое расписание более лучшим. Помимо целевой страницы на веб-сайте были реализованы страницы, такие как: “Блог”, “Вход/Регистрация”.

Для достижения поставленной цели для курсового проекта функционально были реализованы следующие задачи:

1. Анализ аналогичных решений;
2. Выбор способа верстки;
3. Выбор стилевого оформления;
4. Разработка логотипа;
5. Разработка пользовательских элементов;
6. Разработка спецэффектов;
7. Добавление веб-сайту адаптивности;
8. Кроссбраузерность веб-сайта;
9. Создание руководства пользователя;

Список используемых литературных источников представлен в пункте 6. Также дополнительная информация находится в приложениях 1-6.

Таким образом, из выше сказанного, можно сделать вывод, что в курсовом проекте раскрывается все теоретические и практические аспекты темы и достигнуты поставленные цели и задачи роботы.

# Список использованных источников

1. Документация по JavaScript/HTML [Электронный ресурс]. - Режим доступа : <https://developer.mozilla.org>. – Дата обращения 02.05.2022.
2. Документация по Figma [Электронный ресурс]. - Режим доступа: <https://help.figma.com>. – Дата обращения 02.05.2022.
3. Документация по Prepros [Электронный ресурс]. - Режим доступа : <https://prepros.io/help/getting-started>. – Дата обращения 02.05.2022.
4. Документация по Sass/Scss [Электронный ресурс]. - Режим доступа : <https://sass-scss.ru/guide/> – Дата обращения 02.05.2022.

# Приложение

## Приложение A Прототипы веб-страниц

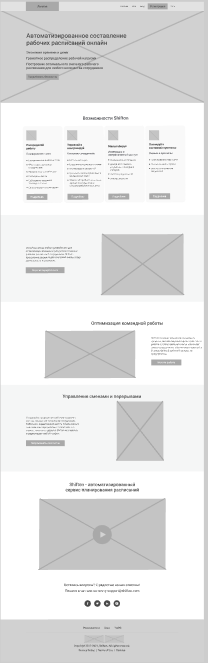
****

Рисунок 1-прототипы страницы “Главная”

## Приложение Б Макет структуры веб-сайта

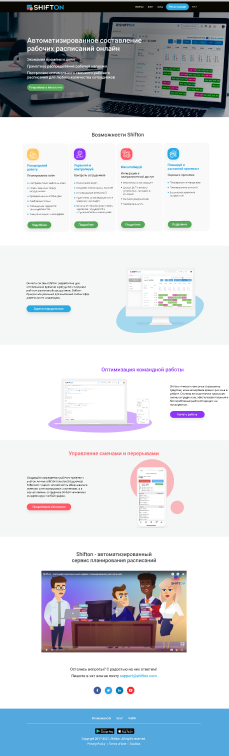
 

Рисунок 1-макет страницы “Главная”

## Приложение В Листинг HTML-документа

|  |
| --- |
| <!-- header -->     <header class="header">...     </header>        <!-- main -->        <main class="main">...        </main>        <!--section -->        <section class="section\_opportunities">...        </section>        <!-- article -->        <article class="article\_one">...        </article>        <article class="article\_two">...        </article>        <article class="article\_tree">...        </article>     <!-- footer -->     <footer>...     </footer> |

Структура HTML—документа

## Приложение Г Листинг (CSS)

|  |
| --- |
| .header {    z-index: 10;    position: fixed;    left: 0;    top: 0;    width: 100%;    background: black;  }  .header .header\_blocks {    height: 60px;    display: flex;    justify-content: space-between;    align-items: center;  }  .header .header\_blocks .header\_menu {    display: flex;    align-items: center;  }  .header .header\_blocks .header\_menu .menu {    list-style: none;    display: flex;  }  .header .header\_blocks .header\_menu .menu li {    margin-right: 20px;  }  .header .header\_blocks .header\_menu .menu li .active\_menu\_main {    color: white;    font-weight: bold;  }  .header .header\_blocks .header\_menu .menu li a {    cursor: pointer;    text-decoration: none;    color: white;  }  .header .header\_blocks .header\_menu .menu\_registration {    padding: 10px 20px;    color: white;    margin: 0 10px;    border-radius: 30px;    background-color: #29aae3;    text-decoration: none;    transition: 0.6s;  }  .header .header\_blocks .header\_menu .menu\_registration:hover {    color: #29aae3;    outline: 1px solid #29aae3;    background-color: white;  }  .header .header\_blocks .header\_menu .choise\_language {    background: none;    color: white;  }  .header .header\_blocks .header\_menu .choise\_language option {    color: black;  } |

|  |
| --- |
| @media (max-width: 1000px) {    .article\_one .container {      overflow: visible;    }    .article\_one .container .wrapper {      padding: 20px 0;    }    .article\_two .container {      width: 100%;      overflow: visible;    }    .article\_tree .container .wrapper {      padding: 20px 0;    }    .hide\_forms .modal-content {      width: 80%;    }    iframe {      height: 400px;    }    .wrapper {      max-width: 900px;    }    .main .main\_\_osnova\_\_info {      text-align: center;      max-width: 100%;      color: white;    }    .section\_opportunities .blocks\_opportunities {      display: grid;      grid-template-columns: repeat(2, 1fr);      gap: 40px;    }    .header .header\_blocks .header\_menu .menu {      display: none;    }    .article\_tree .container .create\_schedule .photo\_schedule img {      display: inline-block;      margin-left: 0px;    }    .header .header\_blocks .header\_menu .menu\_registration {      display: none;    }    .regi {      display: flex;    }    .hide {      display: block;    }    .main .main\_\_osnova\_\_info .btn {      font-size: 20px;    }    .sign .modal-content {      width: 90%;    }    .hide\_forms .modal-content {      width: 90%;    }  }  @media (max-width: 515px) {...  }  @media (max-width: 340px) {...  }  @media (max-width: 300px) {...  }  @media (max-width: 260px) {...  } |
| Медиа-запросы |

## Приложение Д Листинг XML

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <root>     <string>        <id>1</id>        <title>Как решить распространенные проблемы с составлением рабочих расписаний</title>        <description>Предлагаем вам ознакомиться с тремя наиболее распространенными проблемами, которые возникают присоздании рабочего расписания, и их оптимальными решениями. Начните внедрять уже сегодня!</description>        <img>Pictures/article/articleFirst.jpeg</img>        <alt>landing</alt>     </string>     <string>        <id>2</id>        <title>5 интересных идей для повышения эффективности работы сотрудников колл-центра</title>        <description>Большинство руководителей колл-центров используют стандартные методы повышения мотивации сотрудников и достижения поставленных целей (например, премии, надбавки и соцпакеты). Но если вы действительно заинтересованы в развитии своей компании, попробуйте новые решения, чтобы повысить производительность операторов.</description>        <img>Pictures/article/articleTwo.jpg</img>        <alt>girl</alt>     </string>     <string>        <id>3</id>        <title>Идеальное рабочее расписание для компании</title>        <description>Планирование намного важнее, чем вы думаете! В этой статье мы расскажем, как составить рабочее расписание, которое будет соответствовать потребностям каждого сотрудника. В долгосрочной перспективе это может существенно увеличить доход вашей компании.</description>        <img>Pictures/article/articleTree.png</img>        <alt>calendar</alt>     </string>  </root> |
| Данные, представленные в XML-формате |

## Приложение Е Листинг SVG

|  |
| --- |
| <svg width="120" height="120" viewBox="0 0 120 120" fill="none" xmlns="http://www.w3.org/2000/svg">  <g filter="url(#filter0\_d\_311\_621)">  <rect x="20" y="16" width="80" height="80" rx="30" fill="#FFC65D"/>  <path d="M49.2184 29.5219C48.2705 29.8032 47.5934 30.1886 46.9476 30.8344C46.2809  .9594Z" fill="white"/>  </g>  <defs>  <filter id="filter0\_d\_311\_621" x="0" y="0" width="120" height="120" filterUnits="userSpaceOnUse" color-interpolation-filters="sRGB">  <feFlood flood-opacity="0" result="BackgroundImageFix"/>  <feColorMatrix in="SourceAlpha" type="matrix" values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 127 0" result="hardAlpha"/>  </filter>  </defs>  </svg> |
| Одна из иконок |

## Приложение Ж Листинг JavaScript Slider

|  |
| --- |
| (function Slider() {    let slide = document.querySelectorAll(".slide");    let right = document.querySelector(".right-btn");    let left = document.querySelector(".left-btn");    let counter = 0;    var slider\_const = 2;    const info\_schedule = document.querySelectorAll(".info\_schedule");    const photo\_schedule = document.querySelectorAll(".photo\_schedule");    const animateH1 = document.querySelectorAll(".h1\_animate");    const section\_opportunities = document.querySelector(      ".section\_opportunities"    );    if (window.screen.width < 1024) {      for (let i = 0; i < info\_schedule.length; i++) {        info\_schedule[i].classList.add("nonActive");        photo\_schedule[i].classList.add("nonActive");        animateH1[i].classList.add("nonActive");      }      section\_opportunities.classList.add("active");    }    if (window.screen.width > 698 && window.screen.width <= 999) {      slider\_const = 1;    } else if (window.screen.width < 697) {      slider\_const = 0;    } else {      slider\_const = 2;    }    window.onresize = () => {      if (window.screen.width > 698 && window.screen.width <= 999) {        slider\_const = 1;      } else if (window.screen.width < 697) {        slider\_const = 0;      } else {        slider\_const = 2;      }      if (window.screen.width < 1024) {        for (let i = 0; i < info\_schedule.length; i++) {          info\_schedule[i].classList.add("active");          photo\_schedule[i].classList.add("active");          animateH1[i].classList.add("active");        }        section\_opportunities.classList.add("active");      }    };    // перетаскивание слайдов по пальцам свайпов    right.onclick = () => {      counter++;      for (let i = 0; i < slide.length; i++) {        if (counter >= slide.length - slider\_const) {          slide[i].style.left = 0 + "px";          counter = 0;        }        slide[i].style.left = -(350 \* counter) + "px";      }    };    left.onclick = () => {      counter--;      for (let i = 0; i < slide.length; i++) {        if (counter < 0) {          slide[i].style.left = 0 + "px";          counter = slide.length - 3;        }        slide[i].style.left = -(350 \* counter) + "px";      }    };    // touch swipe    let startX = 0;    let isBool = false;    let endX = 0;    const slider = document.querySelector(".slider");    slider.addEventListener("touchstart", (e) => {      startX = e.changedTouches[0].clientX;      isBool = true;    });    slider.addEventListener("touchmove", (e) => {      if (isBool) {        endX = e.changedTouches[0].clientX;        if (endX - startX > 0) {          left.click();        }        if (endX - startX < 0) {          right.click();        }        isBool = false;      }    });    slider.addEventListener("touchend", (e) => {      isBool = false;    });    slider.addEventListener("mousedown", (e) => {      startX = e.clientX;      isBool = true;    });    slider.addEventListener("mousemove", (e) => {      if (isBool) {        endX = e.clientX;        if (endX - startX > 0) {          left.click();        }        if (endX - startX < 0) {          right.click();        }        isBool = false;      }    });    slider.addEventListener("mouseup", (e) => {      isBool = false;    });  })();  const info\_schedule = document.querySelectorAll(".info\_schedule");  const photo\_schedule = document.querySelectorAll(".photo\_schedule");  const animateH1 = document.querySelectorAll(".h1\_animate");  const section\_opportunities = document.querySelector(".section\_opportunities");  if (window.screen.width < 1024) {    for (let i = 0; i < info\_schedule.length; i++) {      info\_schedule[i].classList.add("nonActive");      photo\_schedule[i].classList.add("nonActive");      animateH1[i].classList.add("nonActive");    }    section\_opportunities.classList.add("active");  } |
| slider.js — скрипт для слайдера |

## Приложение З Листинг JavaScript Animation

|  |
| --- |
| //animation blocks  (function Animate() {    const info\_schedule = document.querySelectorAll(".info\_schedule");    const photo\_schedule = document.querySelectorAll(".photo\_schedule");    const animateH1 = document.querySelectorAll(".h1\_animate");    const section\_opportunities = document.querySelector(      ".section\_opportunities"    );    const funAnimate = () => {      section\_opportunities.classList.add("active");    };    const FunctionAnimate = (info, photo, text) => {      info.classList.add("active");      photo.classList.add("active");      text.classList.add("active");    };    const scroll\_up = document.querySelector(".scroll");    scroll\_up.addEventListener("click", fun1);    window.onscroll = () => {      if (window.screen.width >= 1024) {        if (          document.documentElement.scrollTop > 1200 &&          document.documentElement.scrollTop < 1650        ) {          FunctionAnimate(info\_schedule[0], photo\_schedule[0], animateH1[0]);        }        if (          document.documentElement.scrollTop > 415 &&          document.documentElement.scrollTop < 815        ) {          funAnimate();        }        if (          document.documentElement.scrollTop > 1900 &&          document.documentElement.scrollTop < 2300        ) {          FunctionAnimate(info\_schedule[1], photo\_schedule[1], animateH1[1]);        }        if (          document.documentElement.scrollTop > 2400 &&          document.documentElement.scrollTop < 2800        ) {          FunctionAnimate(info\_schedule[2], photo\_schedule[2], animateH1[2]);        }      }      if (document.documentElement.scrollTop > 200)        scroll\_up.style.display = "block";      else {        scroll\_up.style.display = "none";      }    };    function fun1() {      window.scrollTo({        top: 0,        behavior: "smooth",      });    }  })();  let taba = document.querySelectorAll(".tablinks");  let tabs\_info = document.querySelectorAll(".tabcontent");  for (let i = 0; i < taba.length; i++) {    taba[i].addEventListener("click", () => {      for (let i = 0; i < taba.length; i++) {        tabs\_info[i].classList.remove("active");        taba[i].classList.remove("active");      }      tabs\_info[i].classList.add("active");      taba[i].classList.add("active");    });  } |
| animate.js — скрипт для анимации блоков |

## Приложение К Листинг JavaScript Burger-menu

|  |
| --- |
| (function Home() {    const hide = document.getElementById("click");    const hidden\_menu = document.querySelector(".hidden\_menu");    const sign = document.querySelector(".sign");    const detailed = document.querySelector(".detailed");    hide.onclick = () => {      if (hidden\_menu.classList.contains("active")) {        hidden\_menu.classList.remove("active");        hide.classList.remove("change");      } else {        hidden\_menu.classList.add("active");        hide.classList.add("change");        hidden\_menu.classList.add("active");      }    };    const menu\_registration1 = document.querySelectorAll(".menu\_registration1");    const close = document.querySelector(".close");    const close\_one\_form = document.querySelector(".close\_one\_form");    const hide\_forms = document.querySelector(".hide\_forms");    const menu\_registration = document.querySelector(".menu\_registration");    for (let i = 0; i < menu\_registration1.length; i++) {      menu\_registration1[i].addEventListener("click", () => {        hide\_forms.style.display = "block";        sign.style.display = "none";      });    }    menu\_registration.addEventListener("click", () => {      hide\_forms.style.display = "block";    });    close.onclick = () => {      hide\_forms.style.display = "none";    };    close\_one\_form.onclick = () => {      hide\_forms.style.display = "none";    };    window.onclick = function (event) {      if (event.target === hide\_forms) {        hide\_forms.style.display = "none";      } else if (event.target === sign) {        sign.style.display = "none";      } else if (event.target === detailed) {        detailed.style.display = "none";        document.body.style.overflow = "visible";      }    };    const mobile\_registration = document.querySelector(".mobile\_registration");    mobile\_registration.onclick = () => {      hide\_forms.style.display = "block";      hidden\_menu.classList.remove("active");      hide.classList.remove("change");    };    setInterval(() => {      if (window.screen.width > 999) {        hidden\_menu.classList.remove("active");        hide.classList.remove("change");      }    }, 0);    const closing\_form2 = document.querySelector(".closing\_form2");    const close\_sign = document.querySelector(".close-sign");    const register\_sigb = document.querySelector(".register\_sigb");    register\_sigb.onclick = () => {      sign.style.display = "block";      hide\_forms.style.display = "none";    };    close\_sign.addEventListener("click", () => {      sign.style.display = "none";    });    closing\_form2.addEventListener("click", () => {      sign.style.display = "none";    });    const hide\_slider = document.querySelector(".hide-slider");    const all\_display\_none = document.querySelector(".all\_display\_none");    const open\_slider = document.querySelectorAll(".open\_slider");    const close\_slider = document.querySelectorAll(".close\_slider");    for (let i = 0; i < close\_slider.length; i++) {      close\_slider[i].onclick = () => {        hidden\_menu.classList.remove("active");        hide.classList.remove("change");        hide\_slider.style.display = "none";        all\_display\_none.style.display = "block";        section\_infomation.style.display = "none";        window.scrollTo({          top: 0,        });      };    }    for (let i = 0; i < open\_slider.length; i++) {      open\_slider[i].onclick = () => {        setTimeout(() => {          window.scrollTo({            top: 0,          });        }, 0);        hide\_slider.style.display = "block";        all\_display\_none.style.display = "none";        section\_infomation.style.display = "none";      };    }    const close\_detailed = document.querySelector(".close\_detailed");    const open\_app = document.querySelectorAll(".open\_app");    for (let i = 0; i < open\_app.length; i++) {      open\_app[i].onclick = () => {        detailed.style.display = "block";        document.body.style.overflow = "hidden";      };    }    close\_detailed.onclick = () => {      detailed.style.display = "none";      document.body.style.overflow = "visible";    };    const section\_infomation = document.querySelector(".section\_infomation");    const open\_blog = document.querySelectorAll(".open\_blog");    for (let i = 0; i < open\_blog.length; i++) {      open\_blog[i].onclick = () => {        hide\_slider.style.display = "none";        hidden\_menu.classList.remove("active");        hide.classList.remove("change");        section\_infomation.style.display = "block";        all\_display\_none.style.display = "none";        window.scrollTo({          top: 0,        });      };    }    const sign\_btn = document.querySelectorAll(".sign-btn");    for (let i = 0; i < sign\_btn.length; i++) {      sign\_btn[i].onclick = () => {        hidden\_menu.classList.remove("active");        hide.classList.remove("change");        sign.style.display = "block";      };    }  })(); |

home.js — скрипт для бургер-меню