**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc168054133)

[1 Создание технического задания 4](#_Toc168054134)

[2 Разработка дизайна веб-интерфейса 6](#_Toc168054135)

[3 Создание интерфейса сайта 8](#_Toc168054136)

[3.1 создание верстки и стилизация страниц 8](#_Toc168054137)

[3.2 добавление динамических элементов на сайт 10](#_Toc168054138)

[3.3 создание адаптивного интерфейса 12](#_Toc168054139)

[ЗАКЛЮЧЕНИЕ 16](#_Toc168054140)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc168054141)

# **ВВЕДЕНИЕ**

Специальность 09.02.07. Информационные системы и программирование, является очень востребованной в настоящее время, поскольку, в связи с активным развитием прогресса в области информационных технологий, этой сфере продолжают требоваться новые сотрудники и исследователи.

Целью данного отчета будет практико-теоретическое воспроизведение содержания учебной практики. Вся оценочная документация, отраженная в дневнике по практике и характеристике будет предложена руководителю практики от ОУ в виде отдельных документов.

В качестве средств документирования практико-теоретических результатов УП будут использоваться скриншоты и фотографии с пояснениями различных этапов выполнения работ. Поскольку для выполнения заданий необходимо уделять внимание теоретической составляющей — формированию основных понятий и представлений, связанных с предметной областью, в отчет будет включена информация из дополнительных источников. Результаты работы будут ежедневно фиксироваться в дневнике по практике и сопровождаться оценкой руководителя практики, в отчете эти данные представлены не будут.

# **1 Создание технического задания**

Выбранная в рамках данной учебной практики предметная область – компания по продвижения и популяризации профессий в области туризма и гостеприимства.

Целевой аудиторией сайта будут школьники и студенты, заинтересованные в работе в этой области; преподаватели соответствующих специальностей; бюджетные и коммерческие, в том числе образовательные, организации, ведущие деятельность в этой области.

Данный сайт выполняет информационную функцию, представляя посетителям информацию о самой организации, сотрудниках, функция, партнерах и т.д. Также на нем можно будет ознакомиться с мероприятиями организации и партнеров и зарегистрироваться на них.

В качестве референсов для дизайна был взят сайт jug.ru, продвигающий конференции для IT-специалистов, и leader-id.ru, сайт для продвижения различных мероприятий и сообществ в области работы и учебы. Примеры карточек с этих сайтов можно увидеть на рисунке 1 и рисунке 2.

Стиль сайта можно описать как современный, деловой. В цветовой гамме использовались белый, свело-голубой и светло-зеленый. Они создают ощущение чистоты, свежести, а также надежности, безопасности и комфорта. Шрифт использовался JetBrains Mono, в нем символы имеют одинаковую ширину, а строчные буквы максимально высокие. Все это позволяет отображать текст четче.

Все вышеперечисленные аспекты, а также подробное описание навигации и содержания страниц были отражены в техническом задании на данную практику.

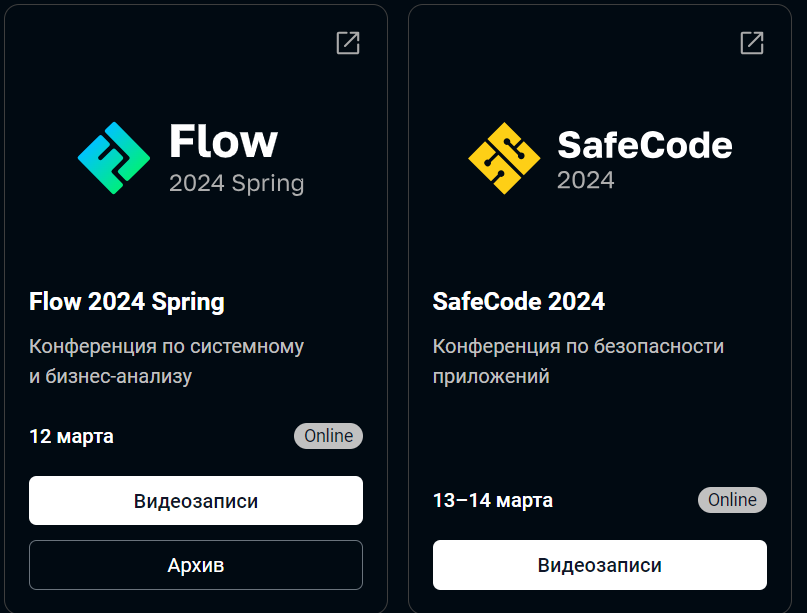


Рисунок 1 - Пример карточки мероприятия с jug.ru

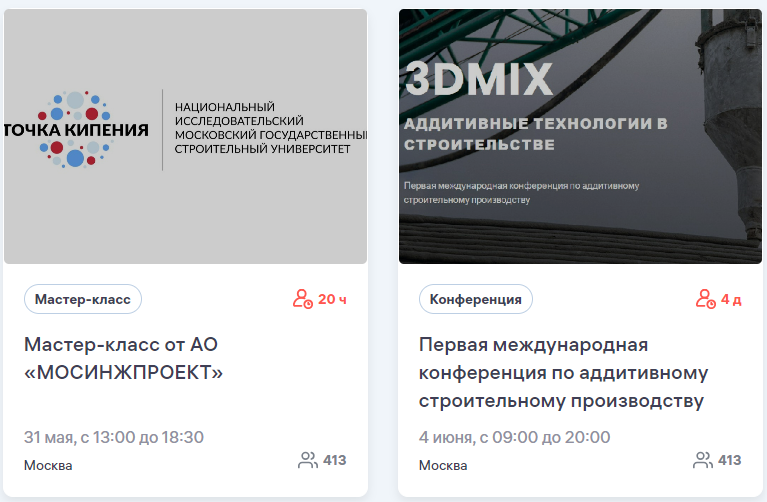


Рисунок 2 – Пример карточки мероприятия с leader-id.ru

# **2 Разработка дизайна веб-интерфейса**

Прежде чем делать дизайн-макет веб-интерфейса, разрабатывается его прототип, по сути, эскиз будущего сайта. Он отображает структуру, расположение и функциональные характеристики элементов, сценарий взаимодействия с пользователем, но без подбора картинок, форм, цветов и шрифтов. Цель прототипа — понять, как будет выглядеть и работать сайт в кратчайшие сроки, чтобы не допускать ошибок проектирования на более поздних стадиях. Разрабатывается прототип на основе технического задания в программе Figma. На рисунке 3 представлен прототип одной из карточек сайта.

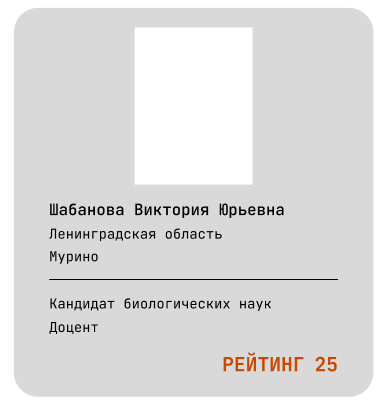


Рисунок 3 – Пример прототипа карточки сайта

После создания прототипа на его основе создается полноценный дизайн-макет, который будет реализовывать разработчик. Он может отходить от решений, принятых на этапе прототипирования, когда в процессе создания дизайна приходят более удачные идеи реализации интерфейса.

В данной работе был создан прототип средней степени проработанности, в нем отражены элементы сайта, шрифты и пункты меню.

При разработке дизайна были изменены размеры некоторых надписей, формы и размеры некоторых элементов, подобраны цвета, добавлены новые элементы, подобраны иллюстрирующие пункты меню картинки, которые имели бы одну стилистику.

В качестве базового цвета был выбран светло-зеленый #F9FFF9 – цвет карточек. Акцентом будут ярко-синий #009FFC и ярко-зеленый #419A1F, они помогут выделить смысловые части, основной контент на странице и функциональные элементы, кнопки. Светло-голубой цвет #B9EEFF – цвет меню. Нейтральным цветом будет темно-синий, почти черный #1E1E2F, это цвет текста. Пример окончательного дизайна одной из карточек после применения цветов представлен на рисунке 4.

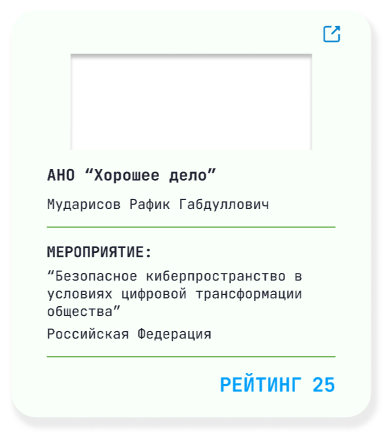


Рисунок 4 – Пример дизайна карточки сайта

# **3 Создание интерфейса сайта**

## **3.1 Создание верстки и стилизация страниц**

Разработка веб-интерфейса на основе созданного дизайна начинается с разметки страниц на составные части с использованием БЭМ.

БЭМ – компонентный подход к веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки. Он позволяет легко и быстро разрабатывать интерфейсы любой сложности и повторно использовать существующий код.

На рисунке 5 представлена HTML-разметка «подвала» разрабатываемого сайта. А на рисунке 6 представлен результат отображения HTML-кода в браузере.

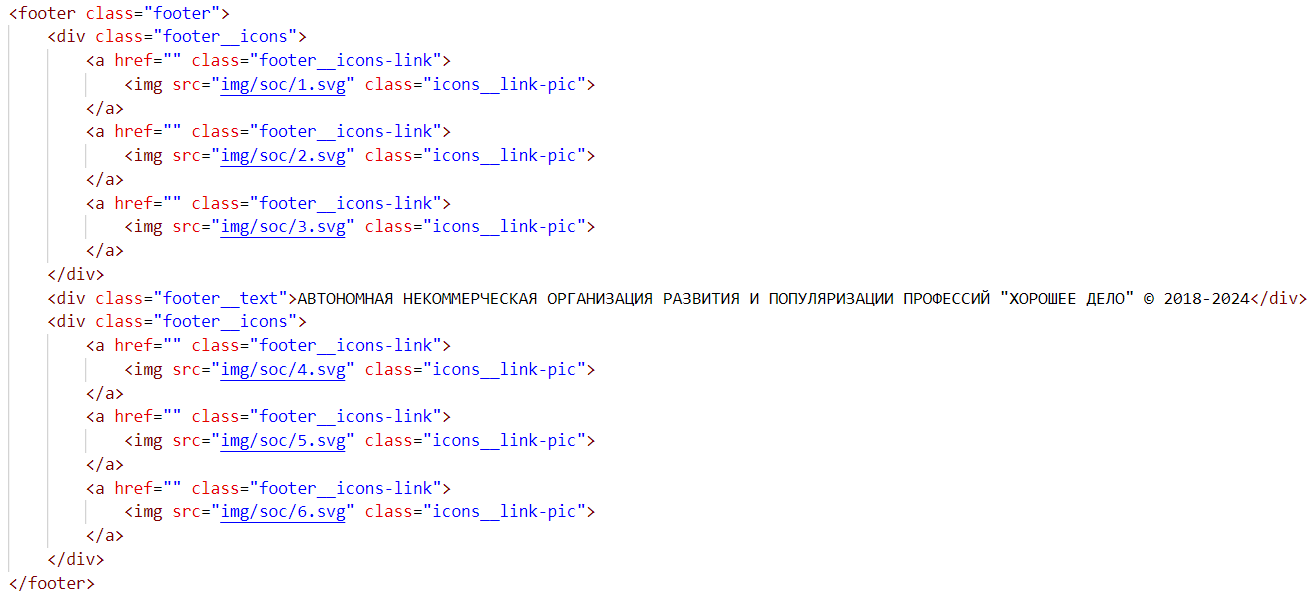


Рисунок 5 – HTML-код «подвала» сайта

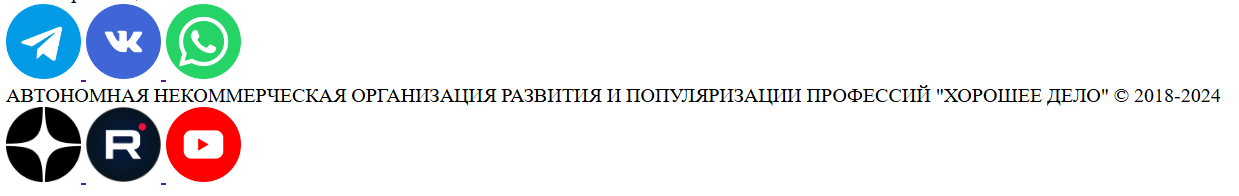


Рисунок 6 – Отображение «подвала» сайта в браузере

После создания HTML-разметки страницы сайта стилизуются с помощью SCSS-кода, который преобразуется в таблицы стилей CSS. На рисунке 7 показана стилизация с помощью SCSS «подвала» сайта, результат показан на рисунке 8.

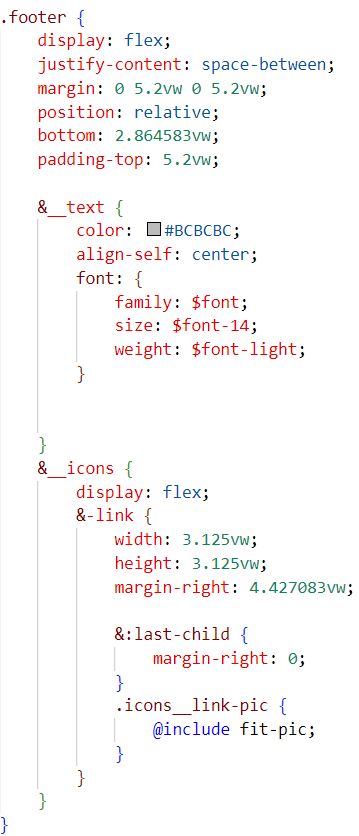


Рисунок 7 – SCSS-код для применения стилей к «подвалу» сайта



Рисунок 8 – Стилизованный «подвал» сайта

## **3.2 Добавление динамических элементов на сайт**

На данном сайте динамическими элементами будут меню, слайдер с новостями и lottie-анимация.

Для создания меню, при наведении на пункты которого будут отображаться подпункты, используется код на языке программирования JavaScript, далее он подключается к HTML-файлу страницы сайта. На рисунке 9 представлен код функции раскрытия подменю на JavaScript.



Рисунок 9 – Код функции раскрытия подменю на JavaScript

Для создания слайдера будет использоваться готовый код из библиотеки Bootstrap. Для этого нужно скопировать и вставить на страницу сайта готовую HTML-верстку, скачать файловую структуру SCSS и JS, подключить необходимые для слайдера файлы к странице сайта, так же как свои собственные файлы, изменить исходный код файлов Bootstrap при необходимости. На рисунке 10 представлен слайдер на данном сайте уже со всеми внесенными изменениями.

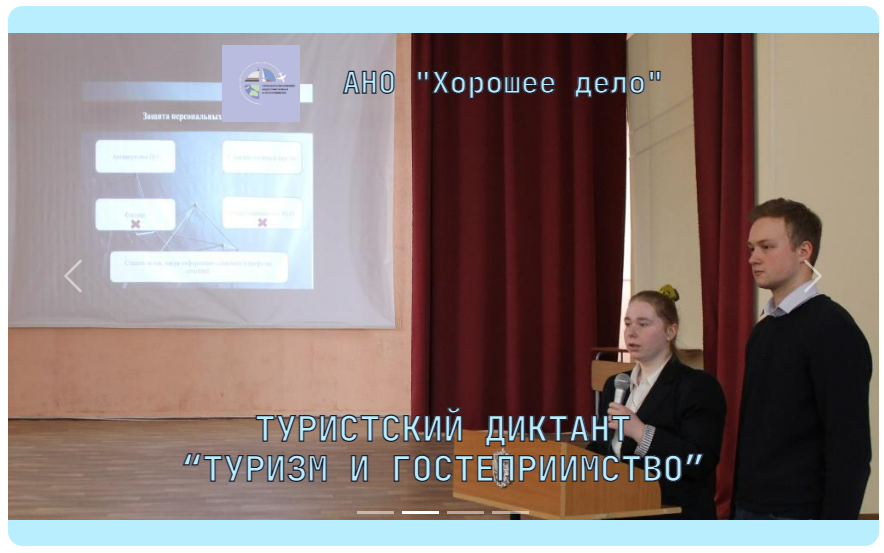


Рисунок 10 – Слайдер на сайте с использованием Bootstrap

Lottie-анимация на сайте будет представлять собой компас с вращающимися стрелками. Для ее создания в программе Figma был нарисован сам компас, анимирован с помощью плагина Aninix UI animation в Figma. Далее он был импортирован и подключен на сайт с помощью специальной библиотеки Lottie-web, которая делает анимацию на сайтах подвижной. На рисунке 11 представлен анимированный элемент сайта.

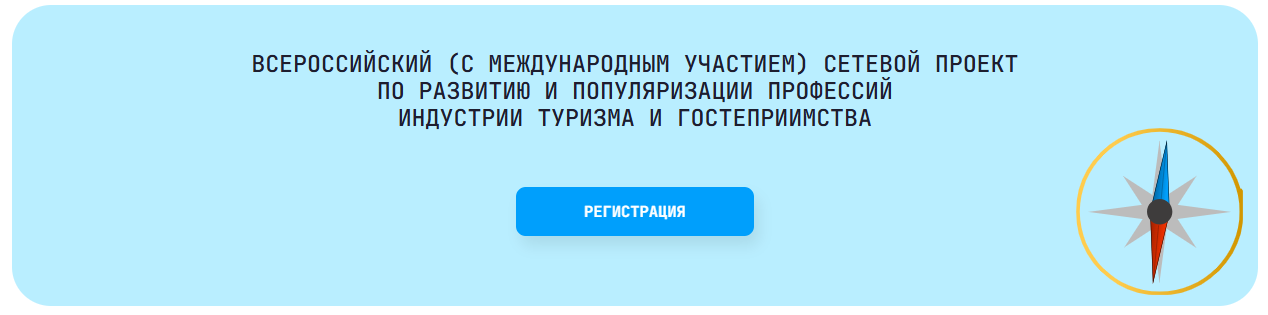


Рисунок 11 – Анимированный компас на сайте

А на рисунке 12 представлен JavaScript-код подключения анимации на сайт с помощью библиотеки Lottie-web.

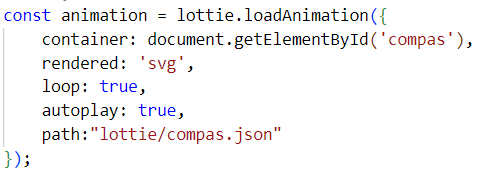


Рисунок 12 – JavaScript-код подключения библиотеки Lottie-web

## **3.3 Создание адаптивного интерфейса**

Далее нужно создать адаптивную верстку, то есть чтобы элементы сайта корректно отображались на устройствах разного размера в соответствии с созданным дизайном планшетной и мобильной версий.

Для этого используется директива @media в CSS и SCSS, она реализует медиа-запрос. Это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств. Внутри медиа-запроса задается ширина экрана устройства в пикселях и стили элементов, которые должны применяться при этой указанной ширине.

В SCSS-коде были созданы медиа-запросы для разрешения экрана 1400px, 1200px, 992px, 768px, 576px, 480px, они используются через директиву @mixin, что является аналогом функции в языках программирования. На рисунке 13 представлен SCSS-код масштабирования кнопки на сайте под разные разрешения.

Особое внимание стоит уделить адаптации бокового меню. В обычном состоянии меню имеет в одном пункте картинку и подпись и занимает от 30% до 40 % ширины экрана. Но при разрешении 768px и меньше, чтобы все было компактнее, в меню остаются только иконки, а подписи к ним убираются. При нажатии или наведении на меню подписи выплывают сбоку.



Рисунок 13 – Адаптация кнопки под разные разрешения экрана

Чтобы создать такой функционал, нужно написать функцию на JavaScript, которая будет отслеживать масштаб экрана и его изменения. Если ширина экрана меньше 768px, то подписи к пунктам меню скрываются. Функция отслеживает движения пользователя по экрану, и при наведении на меню появляются скрытые подписи, а при убирании указателя из области меню они снова скрываются. Код этой функции и ее вызова представлен на рисунке 14.



Рисунок 14 – JavaScript-код масштабирования бокового меню

В результате создания адаптируемого интерфейса сайт стал доступен для просмотра на всех устройствах. На рисунках 15, 16, 17, 18 показано, как отображается главная страница сайта и раскрытое меню на компьютере с разрешением более 1400px, на планшете с горизонтальной ориентацией с разрешением более 992px, на планшете с вертикальной ориентацией с разрешением 768px и на телефоне с разрешением 576px соответственно.

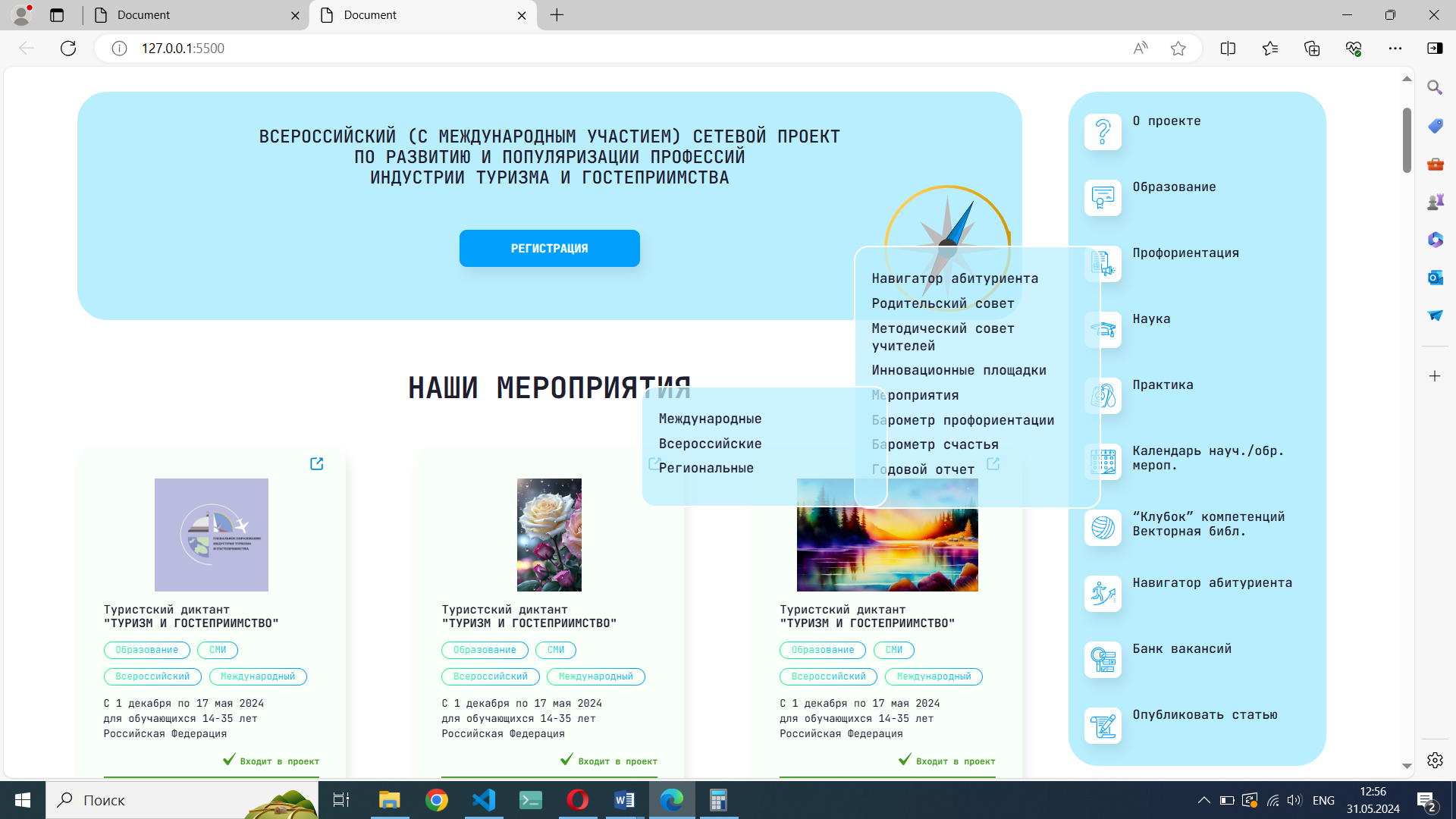


Рисунок 15 – Отображение на компьютере с разрешением более 1400px

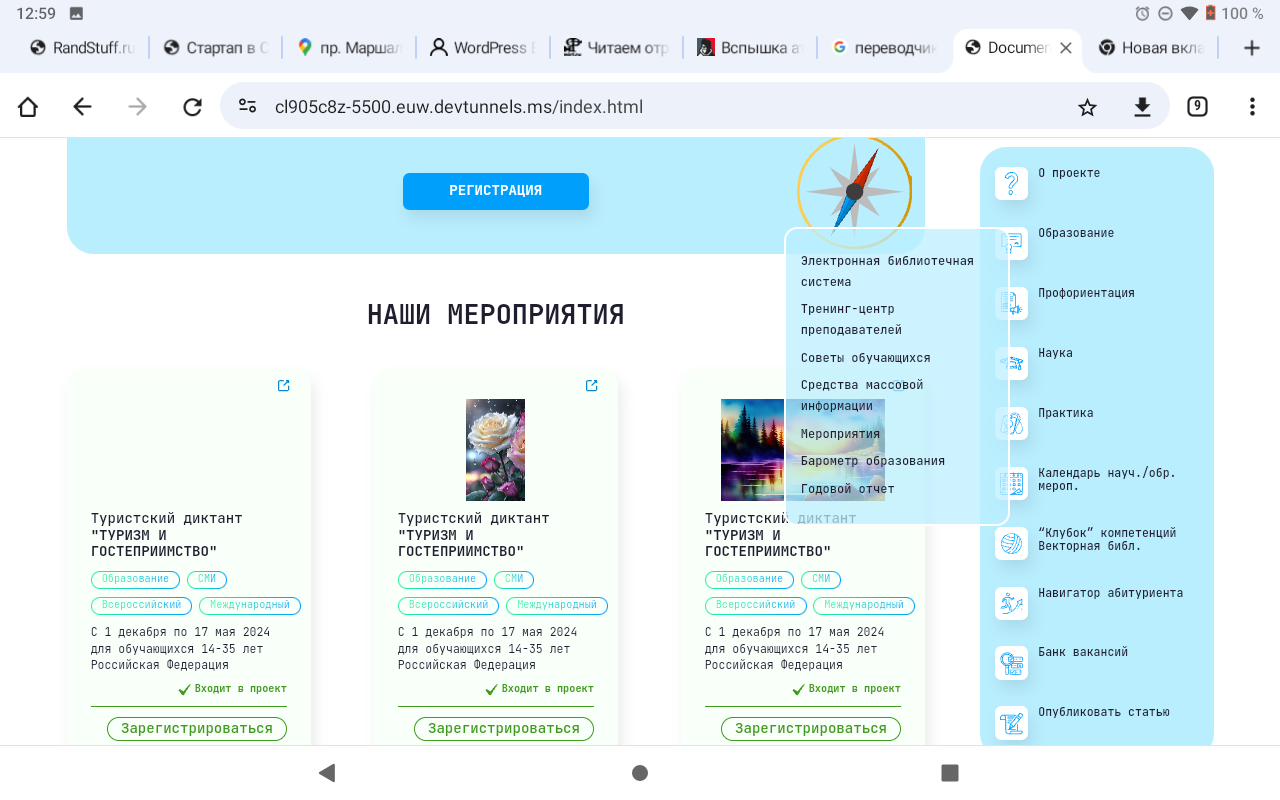


Рисунок 16 – Отображение на планшете с разрешением более 992px

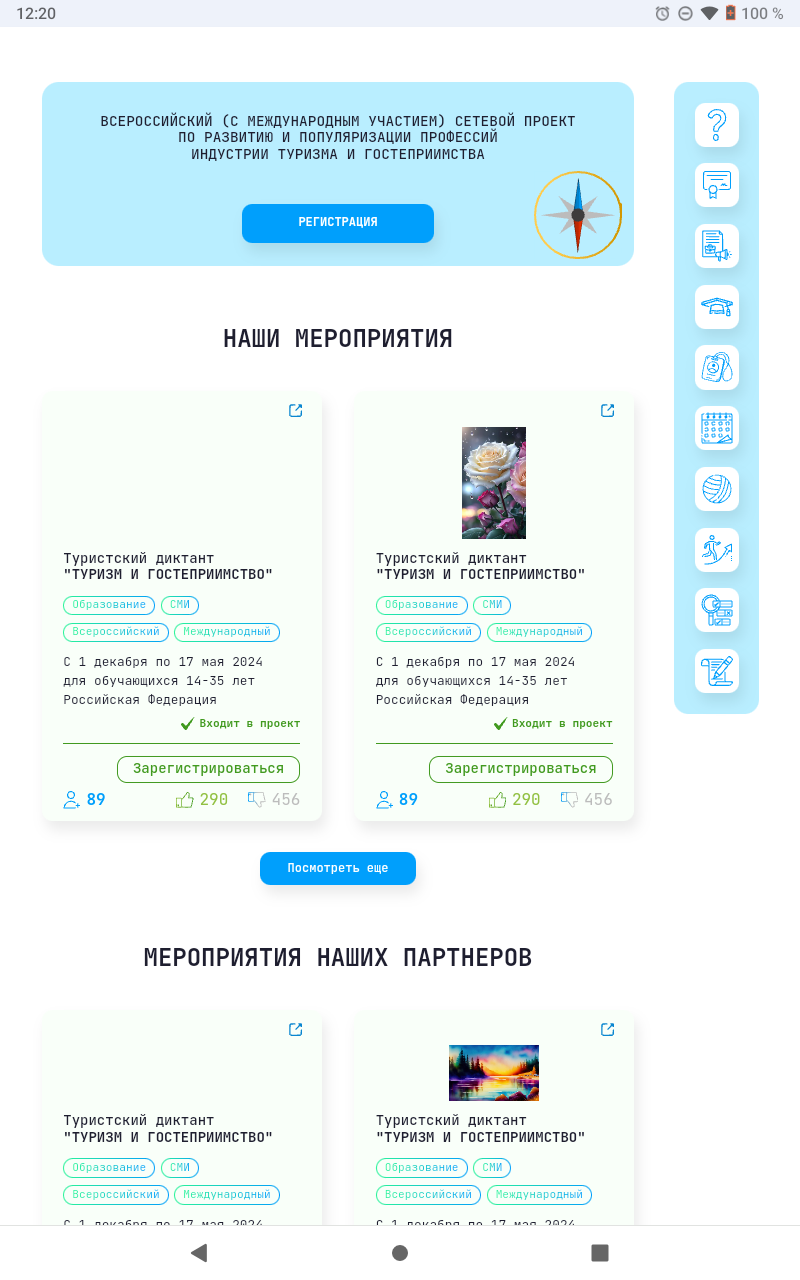


Рисунок 17 – Отображение на планшете с разрешением менее 768px

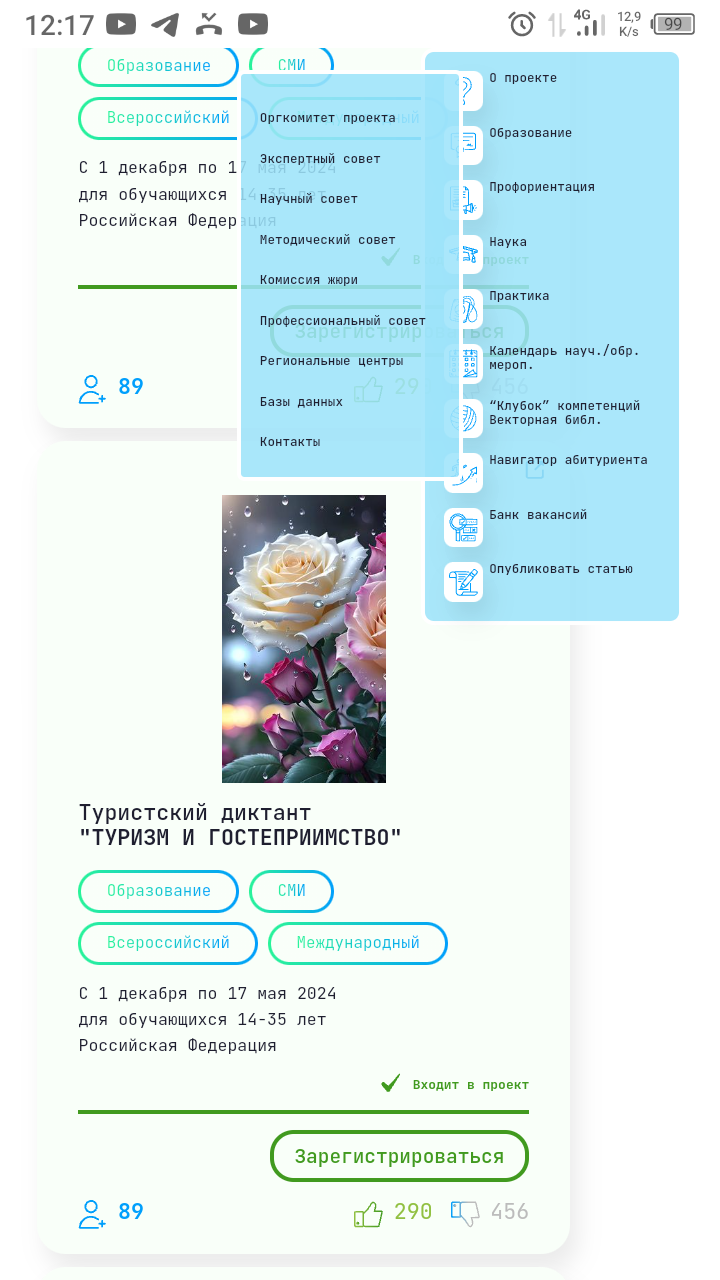


Рисунок 18 – Отображение на телефоне с разрешением менее 576px

# **ЗАКЛЮЧЕНИЕ**

Таким образом, за время прохождения учебной практики были освоены все необходимые практические компетенции (ПК 8.1 — ПК 8.3), предполагаемые профессиональным модулем ПМ.08: «Разработка дизайна веб-приложений». В ходе работы мною были приобретены и отточены следующие навыки:

- разработка технического задания на веб-приложение в соответствии с требованиями;

- разработка дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика;

- проектирование дизайна сайта с применением промежуточных эскизов, требований к эргономике в технической эстетике;

- анализ целевой аудитории при разработке сайта;

- формирование требований к дизайну веб-приложений на основе анализа предметной области и целевой аудитории;

- разработка графических макетов для сайта с использованием современных стандартов;

- создание, использование и оптимизация изображений для сайта с использованием специальных графических редакторов;

- создание lottie-анимации к сайту;

- разработка интерфейса пользователя сайта с использованием современных стандартов;

- создание стилевого оформления сайта с помощью каскадных таблиц стилей;

- создание динамических элементов.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Гаевский А.Ю. 100% самоучитель Создание web-страниц и web-сайтов [Текст]/ Гаевский А.Ю., Романовский В.А. – М.: Триумф, 2022. – 464 c.

2. Клеон Остин Просто продолжай. 10 способов оставаться креативным в любые времена [Текст]: пер. с англ. Юлии Змеевой / Клеон Остин. — М. : Манн, Иванов и Фербер, 2020. — 224 c. : ил.

3. Куликов С. С. Тестирование программного обеспечения. Базовый курс [Текст]/ Куликов С. С. — 3-е изд. — Минск: Четыре четверти, 2020. — 312 с.

4. Пулин Ричард Школа дизайна: макет. Практическое руководство для студентов и дизайнеров [Текст]// Ричард Пулин; пер. с англ. Д. Семеновой; науч. ред. Л. Гроздова. — М. : Манн, Иванов и Фербер, 2020. — 232 c. : ил.

5. Снелл Нэд Абсолютно ясно о создании Web-станиц и Web-сайтов [Текст]// Нэд Снелл; пер. с англ. А. Климович. – Москва: Триумф, 2020. – 224 c.

6. Яворски Питер Ловушка для багов. Полевое руководство по веб-хакингу [Текст]/ Яворски Питер. — СПб.: Питер, 2020. — 272 с.: ил. — (Серия «Библиотека программиста»)

7. WHATWG [Электронный ресурс] URL: <https://html.spec.whatwg.org>

Дата обращения: 23.12.2023

8. Sass [Электронный ресурс] URL: <https://sass-scss.ru> Дата обращения: 23.13.2023

9. code.mu [Электронный ресурс] URL: <https://code.mu/ru/> Дата обращения: 23.10.2023

10. Дока [Электронный ресурс] URL: <https://doka.guide> Дата обращения: 23.10.2023

11. HTML5BOOK.RU [Электронный ресурс] URL: <https://html5book.ru> Дата обращения: 20.05.2023

12. Javascript.ru [Электронный ресурс] URL: <https://learn.javascript.ru> Дата обращения: 15.05.2023

13. htmlbase.ru [Электронный ресурс] URL: <https://htmlbase.ru> Дата обращения: 10.05.2023

14. MDN Web Docs [Электронный ресурс] URL: <https://developer.mozilla.org/ru/> Дата обращения: 19.05.2023

15. БЭМ [Электронный ресурс] URL: <https://ru.bem.info/methodology/> Дата обращения: 09.05.2023

16. Эвристики Нильсена [Электронный ресурс]// SIMPLEONE – 12.05.2020 – URL: <https://simpleone.ru/glossary/evristiki-nilsena/> Дата обращения: 02.05.2023

17. Информационная архитектура: руководство для UX-дизайнеров [Электронный ресурс]// UPROCK – URL: <https://www.uprock.ru/articles/informacionnaya-arhitektura-rukovodstvo-dlya-ux-dizaynerov> Дата обращения: 02.05.2023