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 «Разработка структуры веб-сайта прогноза погоды»

**Исполнитель**

студент(ка) 1 курса \_\_ группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Б. Мысливцев

подпись, дата

**Руководитель**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Сазонова

должность, ученая степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Сазонова

подпись дата инициалы и фамилия

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра информационных систем и технологий

Утверждаю

И.о. заведующего кафедрой

\_\_\_\_\_\_\_\_\_\_\_ Е.А. Блинова

подпись инициалы и фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**ЗАДАНИЕ**

**на курсовое проектирование**

по дисциплине«Компьютерные языки разметки»

Курс 1 Группа: 1 1-40 05 01 «Информационные системы и технологии»

Студент: Мысливцев Алексей Борисович

**1. Тема: «Разработка структуры веб-сайта прогноза погоды**»

**2. Срок выполнения курсового проекта** : с 5 февраля 2024 г. по 06 мая 2024 г.

**3. Технические требования :**

3.1 Прототип веб-сайта должен быть разработан с использованием графических редакторов Figma/Adobe XD/Sketch.

3.2 Для хранения данных должен быть использован XML-формат.

3.3 Разметка содержания сайта должна быть выполнена с применением HTML5 и XML.

3.4 Для описания внешнего вида веб-страниц использовать SCSS и CSS3.

3.5 Веб-сайт должен содержать:

– семантические теги HTML5;

– графические элементы в форме SVG;

– несколько веб-страниц;

– JavaScript для управления элементами DOM.

3.6 Верстка сайта должна быть адаптивной и кроссбраузерной;

3.7 Для тестирования использовать The W3C Markup Validation Service/Git Super Linter

3.8 Проект и пояснения к проекту должны быть размещены на GitHub.

**4. Содержание пояснительной записки**

1. Титульный лист;

2. Задание на курсовое проектирование;

3. Введение;

4. Постановка задачи;

5. Проектирование веб-сайта

6. Реализация структуры веб-сайта

7. Тестирование веб-сайта

8. Заключение

9. Список использованных источников

10. Приложения (полный исходный текст программы разработанного приложения с подробными комментариями)

**5. Форма представления на GitHub выполненного курсового проекта:**

– Теоретическая часть пояснительной записки курсового проекта должна быть представлена в формате MS Word.

– Оформление записки должно быть согласно правилам.

– Листинги представляются в приложении.

**Календарный план**

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта |
| 1 | Задание на курсовое проектирование. Титульный лист | 05.02.2024 |
| 2 | Введение. Постановка задачи | 20.02.2024 |
| 3 | Проектирование веб-сайта | 04.03.2024 |
| 4 | Реализация структуры веб-сайта на HTML5 и внешнего оформления на SCSS и СSS3 | 18.03.2024 |
| 5 | Тестирование веб-сайта | 15.04.2024 |
| 6 | Заключение | 23.04.2024 |
| 7 | Список использованных литературных источников. Приложения | 25.04.2024 |
| 6 | Подготовка и оформление пояснительной записки курсового проекта | 30.04.2024 |
| 7 | Сдача на допуск к защите курсового проекта | 06.05.2024 |
| 8 | Защита курсового проекта | 13.05-24.05.2024 |

**5. Дата выдачи задания** «05» февраля 2024 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В.Сазонова

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Б. Мысливцев

(дата и подпись студента)

Содержание

[Введение 5](#_Toc166181346)

[1. Постановка задачи 6](#_Toc166181347)

[**1.1.** **Обзор аналогичных решений** 6](#_Toc166181348)

[**1.2.** **Техническое задание** 7](#_Toc166181349)

[**1.3.** **Выбор средств реализации программного продукта** 8](#_Toc166181350)

[**1.4. Вывод** 9](#_Toc166181351)

[2. Проектирование страниц веб-сайта 10](#_Toc166181352)

[**2.1. Выбор способа верстки** 10](#_Toc166181353)

[**2.2. Выбор стилевого оформления** 10](#_Toc166181354)

[**2.3. Выбор шрифтового оформления** 11](#_Toc166181355)

[**2.4. Разработка пользовательских элементов** 11](#_Toc166181356)

[**2.5. Выводы** 12](#_Toc166181357)

[3. Реализация структуры веб-сайта 13](#_Toc166181358)

[**3.3. Использование стандартов XML (SVG)** 15](#_Toc166181359)

[3.4 Использование JavaScript 19](#_Toc166181360)

[**3.4. Выводы** 20](#_Toc166181361)

[4. Тестирование веб-сайта 21](#_Toc166181362)

[**4.1. Адаптивный дизайн веб-сайта** 21](#_Toc166181363)

[**4.2. Кроссбраузерность веб-сайта** 23](#_Toc166181364)

[**4.3.** **Тестирование кода** 25](#_Toc166181366)

[4.5 Выводы 26](#_Toc166181367)

[Заключение 26](#_Toc166181368)

[6. Список использованных литературных источников. 28](#_Toc166181369)

[7. Приложение (полный исходный текст программы разработанного приложения с подробными комментариями). 29](#_Toc166181370)

[Приложение 1 Макет структуры веб-сайта 29](#_Toc166181372)

[Приложение 2 Листинг НТML-документа 29](#_Toc166181373)

[Приложение 3 Листинг Sass 29](#_Toc166181374)

[Приложение 4 Листинг CSS 29](#_Toc166181375)

[Приложение 5 Листинг XML-файлов 29](#_Toc166181376)

[Приложение 6 Листинг Js 29](#_Toc166181377)

Введение

Целью данного курсового проекта является разработка веб-сайта на тему «Прогноз погоды», который, помимо предоставления прогноза погоды, будет включать информацию о погодных явлениях и обучающие материалы.

Задачи курсовой работы:

1. Проанализировать существующие веб-сайты по похожей тематике;

2. Разработать макет и прототип веб-сайта;

3. Разработать структура веб-сайты;

4. Наполнить сайт текстовой информацией по теме;

5. Протестировать веб-сайт;

6. Разработать руководство пользователя.

Актуальность курсовой работы обусловлена потребностью в надежных прогнозах погоды, а также возрастающим интересом общества к образовательному контенту в мобильных приложениях. Разработка такого приложения позволит не только удовлетворить эти потребности, но и способствует расширению границ обычного понимания приложений для прогноза погоды, превращая их в инструмент для обучения и получения новых знаний.

1. Постановка задачи

Объектом исследования являются инструменты программирования для создания сайта «Прогноз погоды», интеграция с внешними сервисами, и использование геолокации.

## **Обзор аналогичных решений**

Перед тем, как приступить к разработке макета веб-сайта – необходимо изучить уже существующие сайты, выделить их недостатки и преимущества, которые стоит учесть во время работы, чтобы подготовить наиболее качественное решение для пользователей

* + 1. <https://yandex.by/pogoda>

На рисунке 1.1 представлен внешний вид основной страницы сайта [1].

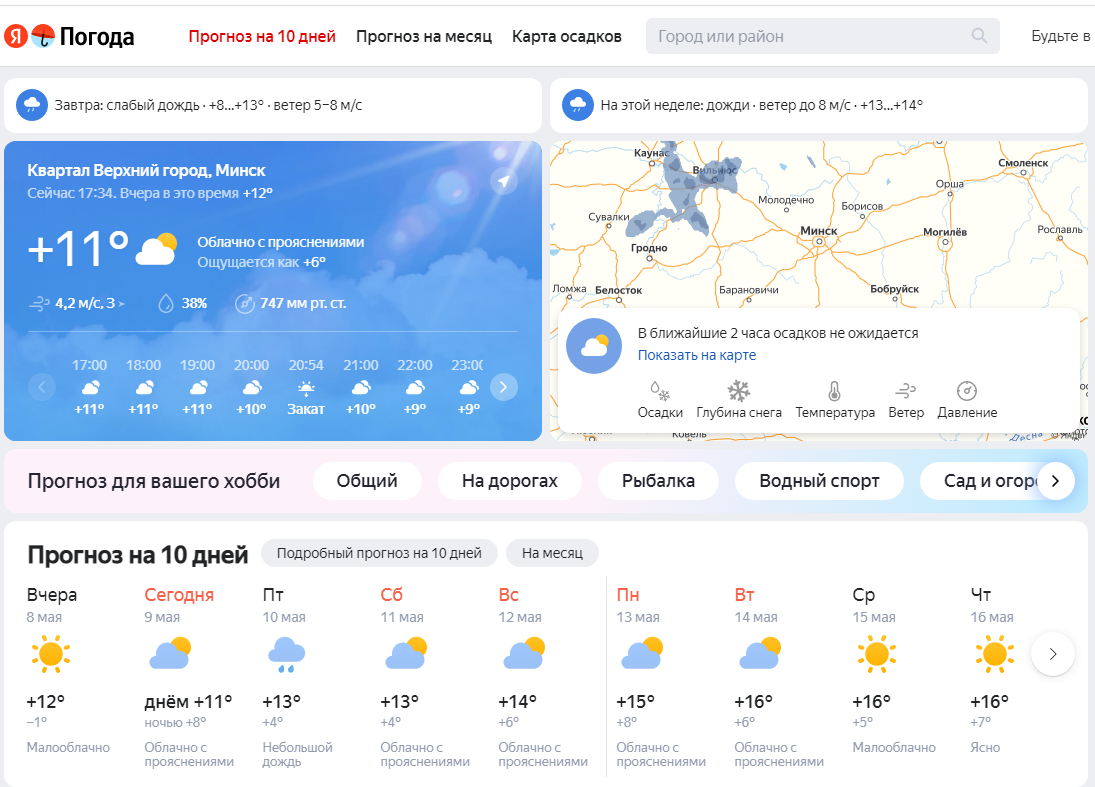


Рисунок 1.1 – Основная страница сайта «Яндекс Погода»

На «Яндекс Погоде» можно найти прогноз погоды на текущий день, 10 и 30 суток. Причем благодаря технологии «Метеум», которая обрабатывает информацию от тысяч приборов на Земле и в космосе, данные обновляются каждый раз, как только пользователь обращается к сервису.

На сайте можно найти: фактическую температуру; показатель «ощущается как»; скорость ветра; осадки; давление; влажность; продолжительность светового дня.

У «Яндекс Погоды» есть удобные приложения на Android и iOS. Там пользователи могут установить уведомление об изменениях погоды. С этой функцией можно вовремя понять, как одеться так, чтобы было комфортно.

Плюсами являются удобный и интуитивно понятный интерфейс сайта и мобильного приложения; прогноз погоды более чем в 200 странах мира; есть карта города, на которой указаны погода и температура в разных его районах.

Минусами являются наличие рекламы, а также то, что новая технология «Метеум» еще обкатывается, поэтому иногда прогноз погоды для конкретного района может быть не совсем верным.

* + 1. [https://nuipogoda.ru](https://nuipogoda.ru/)

На рисунке 1.2. представлен внешний вид основной страницы сайта [1].

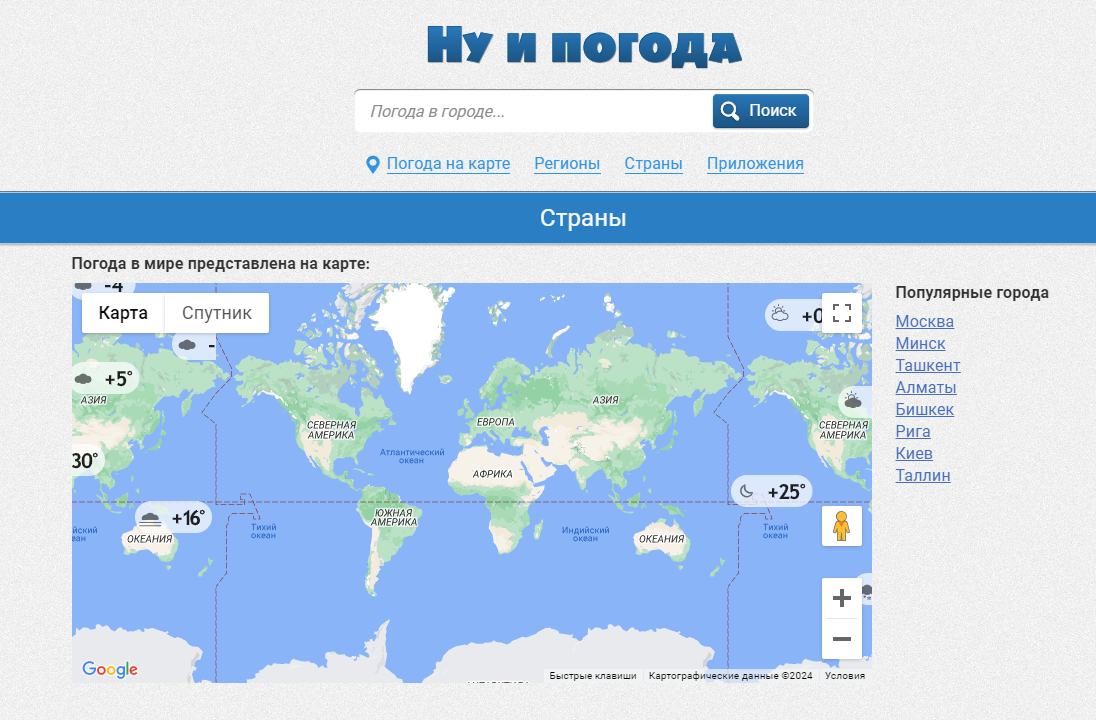


Рисунок 1.2 – Основная страница сайта «Ну и погода»

«Ну и погода» предоставляет собой сервис с поминутным прогнозом погоды. На сайте можно найти информацию на сегодня, завтра, 2 недели и месяц. У сервиса есть приложение на Android и iOS с удобным интерфейсом.

На сайте можно увидеть текущую и фактическую температуру; осадки; давление; влажность; силу ветра.

Плюсами являются простой и понятный интерфейс сайта; краткая сводка с важной информацией представлена на главной странице; есть расширенная версия прогноза погоды; автоматическое определение геолокации пользователя; точный поминутный прогноз погоды.

Минусами являются наличие рекламы, а также то, что не учтен показатель «ощущается как» и вероятность осадков.

## **Техническое задание**

Целью курсового проекта является разработка веб-сайта «Прогноз погоды», целевая аудитория которого очень широкая, так как не один человек не выходит из дома не уточнив погоду на сайте.

В рамках данного проекта требуется создать удобный и интуитивно понятный интерфейс веб-сайта, обеспечивающий лёгкий доступ к различным разделам информации о погоде. Навигация по сайту должна быть простой и понятной для всех категорий пользователей.

Для наглядности и удобства представления информации планируется использовать иллюстрации и изображения, позволяющие лучше визуализировать изучаемые объекты.

Верстка сайта должна быть адаптивной и кроссбраузерной. Важным требованием является создание адаптивного дизайна веб-сайта, который будет корректно отображаться на различных устройствах, включая компьютеры, планшеты и смартфоны. Разработка сайта должна быть ориентирована на обеспечение приятного и удобного пользовательского опыта независимо от используемого устройства.

В соответствии с техническим заданием необходимо также провести тестирование созданного веб-сайта и оптимизировать его работу для обеспечения стабильного функционирования на различных платформах и веб-браузерах.

## **Выбор средств реализации программного продукта**

Проект должен содержать «Разработку структуры веб-сайта прогноза погоды» состоящую из нескольких веб-страниц.

Прототип веб-сайта должен быть разработан с использованием графических редакторов Figma/Adobe XD/Sketch.

Для хранения данных должен быть использован XML-формат.

Разметка содержания сайта должна быть выполнена с применением HTML5 и XML.

Для описания внешнего вида веб-страниц использовать SCSS и CSS3.

В данном подразделе описать выбор языка разметки, инструментов для создания веб-сайта и библиотек, которые будут подключаться

Веб-сайт должен содержать семантические теги HTML5, графические элементы в форме SVG, JavaScript для управления элементами DOM.

Для реализации веб-сайта в целях выполнения курсовой работы был выбран редактор кода Microsoft Visual Studio Code.

При создании веб-сайта были использованы следующие языки: HTML, CSS, JavaScript, XML, а также препроцессор SCSS для CSS.

HTML (HyperText Markup Language) — это стандартизированный язык разметки, используемый для создания структуры веб-страниц. Он состоит из различных элементов, таких как заголовки, списки, изображения и многое другое, которые описывают содержимое страницы и определяют его структуру.

CSS (Cascading Style Sheets) — это язык стилей, который используется для оформления веб-страниц. С помощью CSS можно определять различные аспекты внешнего вида элементов HTML, такие как цвета, шрифты, размеры, расположение и многое другое. CSS позволяет разделять содержимое и представление, что облегчает изменение внешнего вида страницы без изменения её содержимого.

JavaScript — это высокоуровневый язык программирования, который применяется для добавления интерактивности на веб-страницы. Он позволяет создавать динамические элементы на странице, обрабатывать события пользователя, выполнять асинхронные запросы к серверу, анимировать элементы и многое другое.

XML (eXtensible Markup Language): — это расширяемый язык разметки, используемый для хранения и передачи данных. Он представляет собой текстовый формат, состоящий из тегов, которые описывают структуру и содержимое данных. XML часто используется для обмена данными между различными системами, так как он обеспечивает универсальный способ представления информации.

SCSS (Sassy CSS) — это препроцессор CSS, который добавляет некоторые дополнительные функции и возможности к обычному CSS. SCSS позволяет использовать переменные, вложенные правила, миксины, операторы и другие конструкции, которые делают стили более модульными, управляемыми и повторно используемыми.

## **1.4. Вывод**

В рамках данного раздела был проведен анализ существующих готовых решений для создания веб-сайтов, в частности сайтов прогноза погоды. Были выявлены ключевые преимуществ и недостатки в каждом из рассмотренных вариантов.

А также было сформулировано техническое задание, которое определило основное видение проектируемого веб-сайта, задало направление всего проекта. Техническое задание содержит детальное описание функциональных и нефункциональных требований к будущему веб-сайту, а также описывает ожидаемые сценарии использования продукта конечными пользователями.

На основе технического задания был проведен отбор инструментов и технологий для реализации программного продукта. От выбора средств реализации зависит скорость и эффективность разработки, а также удобство последующей поддержки и развития проекта. Выбранные инструменты должны не только обеспечить выполнение всех поставленных задач, но и быть максимально удобными в работе.

2. Проектирование страниц веб-сайта

## **2.1. Выбор способа верстки**

Так как техническое задание включает в себя кроссбраузерность и адаптивность, то за способ верстки была выбрана flexbox вёрстка.

Flexbox (Flexible Box) — это модель макета в CSS, которая позволяет создавать гибкие и адаптивные макеты, основанные на направленном потоке элементов внутри контейнера. С помощью Flexbox можно легко управлять распределением пространства между элементами, их порядком, выравниванием и другими аспектами визуального представления. Flexbox особенно полезен для создания компонентов интерфейса с динамическим содержимым, таких как навигационные меню, списки элементов и блоки с неизвестной шириной или высотой.

## **2.2. Выбор стилевого оформления**

При разработке сайта ключевую роль играет выбор стилевого оформления. Определяется это следующими причинами:

1. Визуальное впечатление: Стиль и дизайн веб-сайта непосредственно влияют на визуальное впечатление посетителей. Привлекательный и профессионально выполненный дизайн может улучшить пользовательский опыт и создать положительное первое впечатление.

2. Удобство использования: Хорошо продуманное стилевое оформление способствует удобству использования сайта. Эффективное использование цветов, шрифтов, размещения элементов и других дизайнерских аспектов помогает создать интуитивно понятный интерфейс, что облегчает навигацию и взаимодействие с контентом.

3. Брендирование: Стилевое оформление сайта может помочь установить и поддерживать брендирование. Использование корпоративных цветов, логотипов, шрифтов и общего стиля помогает создать единое визуальное восприятие, что способствует узнаваемости бренда и повышает доверие к сайту.

4. Адаптивность и доступность: Правильный выбор стилевого оформления также важен для обеспечения адаптивности и доступности сайта. Адаптивный дизайн позволяет сайту корректно отображаться на различных устройствах и разрешениях экранов. При этом учет особых потребностей пользователей с ограниченными возможностями (например, использование доступных цветовых схем для людей с дефицитом зрения) делает сайт более доступным и инклюзивным.

5. SEO-оптимизация: Некоторые аспекты стилевого оформления, такие как правильное использование заголовков, метаданных, альтернативных текстов для изображений и т. д., могут влиять на поисковую оптимизацию (SEO) сайта, улучшая его видимость в поисковых результатах.

## **2.3. Выбор шрифтового оформления**

Выбор шрифтового оформления при разработке веб-сайта имеет несколько важных факторов:

1. Читаемость: плохо выбранный шрифт может затруднить чтения контента и восприятие информации, что негативно скажется на пользовательском опыте.
2. Стиль и атмосфера: шрифты могут помочь создать нужное впечатление о веб-сайте. Например, серьезный бизнес-портал может выбрать классический, профессиональный шрифт, в то время как творческий блог может предпочесть более нестандартный и экспрессивный стиль.
3. Брендинг: хорошо выбранные шрифты могут быть важной частью брендинга. Они могут помочь узнаваемости вашего бренда и передать его уникальный стиль и характер.
4. Совместимость с устройствами: не все шрифты одинаково отображаются на различных устройствах и в разных браузерах. При выборе шрифта следует обращать внимание на его поддержку кроссбраузерности и кроссплатформенности.
5. Производительность: тяжелые шрифты могут замедлить загрузку страницы, особенно на медленных интернет-соединениях или мобильных устройствах. Поэтому важно выбирать шрифты, которые не только красивы, но и эффективны с точки зрения производительности.
6. Доступность: важно также убедиться, что выбранный шрифт доступен для всех пользователей, включая людей с ограниченными возможностями, такими как низкое зрение. Некоторые шрифты могут быть сложными для чтения для людей с дислексией или другими нарушениями чтения.

Учитывая вышеперечисленные аспекты, наиболее подходящим выбором для веб-сайта по данной тематике является семейство шрифтов «Montserrat».

## **2.4. Разработка пользовательских элементов**

В проекте присутствуют элементы, с которыми пользователь может взаимодействовать: навигационное меню, слайдер,

Навигация для веб-сайта — это способ, с помощью которого пользователи перемещаются по веб-сайту. Это включает в себя такие элементы интерфейса, как меню, ссылки, кнопки и другие, помогающие пользователям ориентироваться на сайте и находить необходимую им информацию.

Хорошая навигация делает сайт более удобным для использования, помогает пользователям быстро найти то, что им нужно, и улучшает общий пользовательский опыт. Навигационная структура веб-сайта должна быть логичной и интуитивно понятной, чтобы пользователи могли легко перемещаться между страницами, разделами и функциями сайта.

Навигационное меню данного веб-сайта представлено на рисунке 2.1.

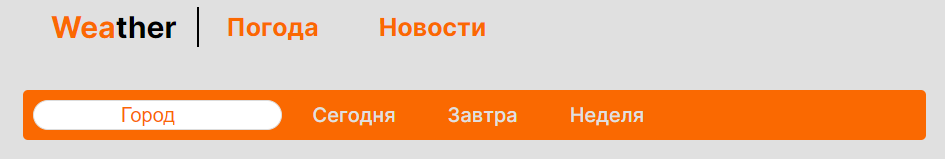


Рисунок 2.1 – Навигационное меню сайта «Прогноз погоды»

Также в проекте присутствуют 3 веб-страницы с непосредственным представлением погоды на сегодня, завтра и неделю.

Изображения данных веб-страниц представлено на рисунках 2.2., 2.3, 2.4

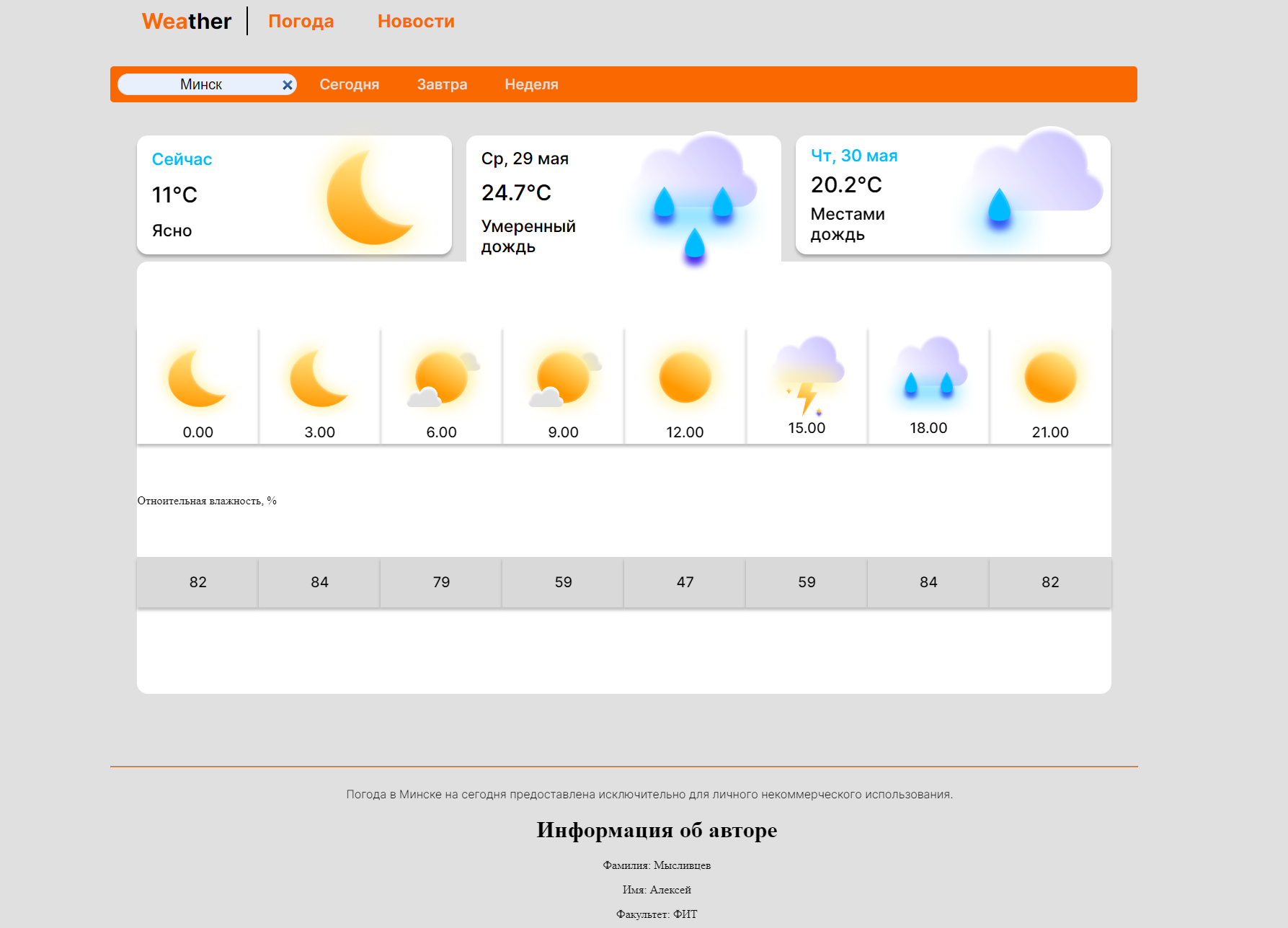


Рисунок 2.2 – Изображение сайта «Прогноз погоды» на сегодня

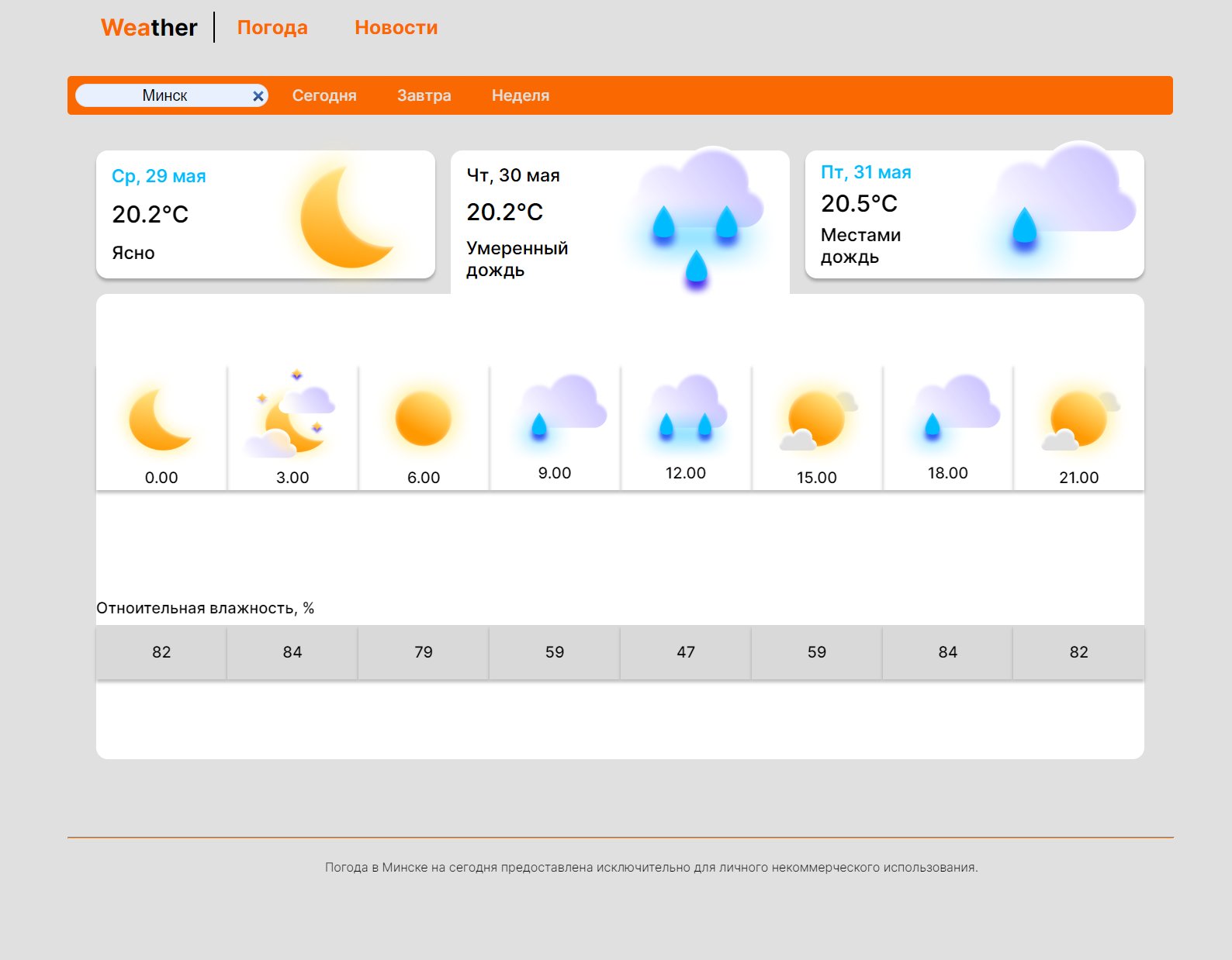


Рисунок 2.3 – Изображение сайта «Прогноз погоды» на завтра

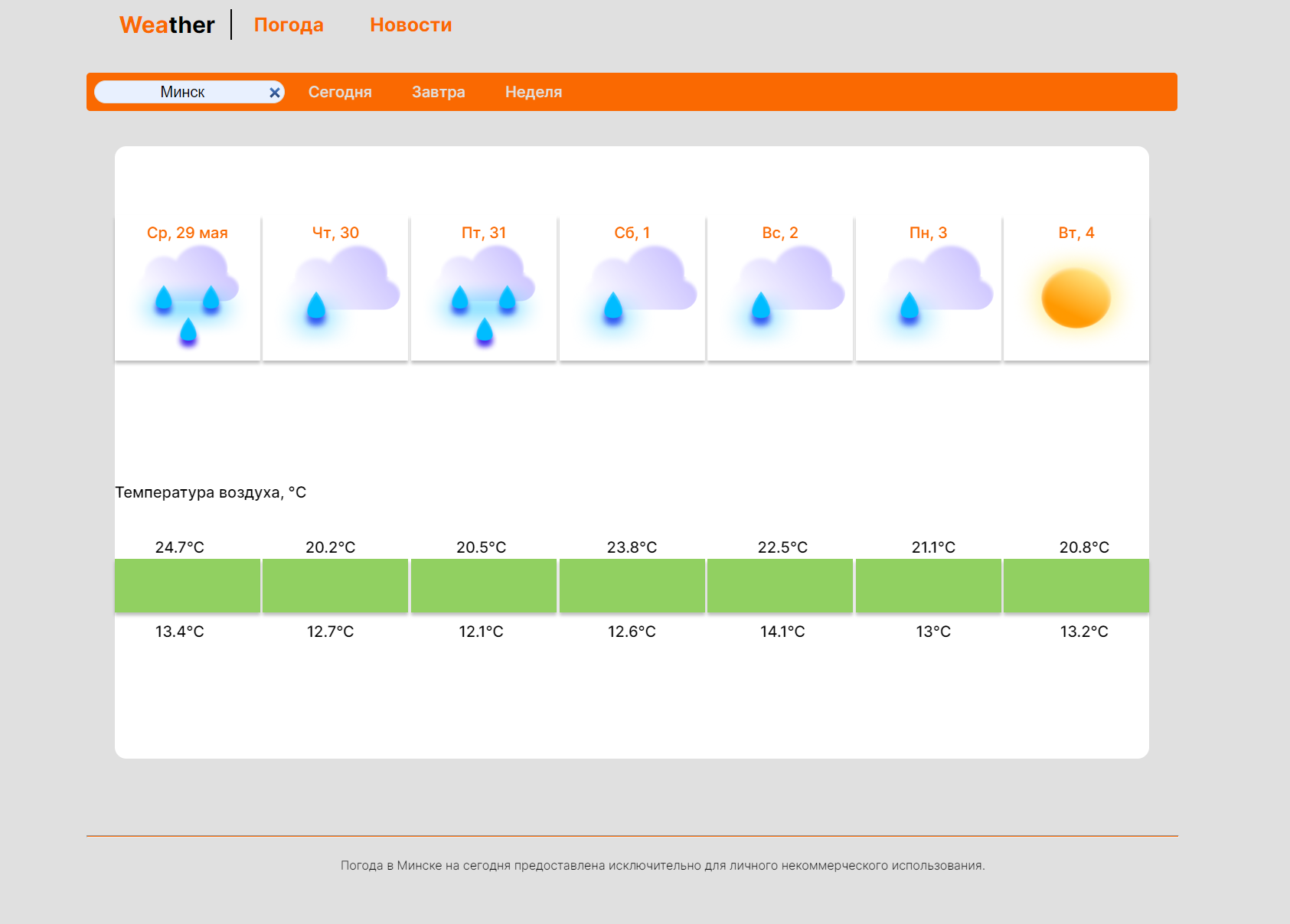


Рисунок 2.4 – Изображение сайта «Прогноз погоды» на неделю

## **2.5. Выводы**

Проектирование страниц веб-сайта является ключевым этапом, определяющим структуру, дизайн и функциональность проекта. Внимание уделяется выбору стилевого оформления, шрифтов, цветовой палитры и элементов дизайна, которые должны соответствовать тематике космоса и делать сайт привлекательным для посетителей.

В данном разделе был спроектирован и создан прототип веб-сайта. Был определен способ верстки, единое стилевое оформление, шрифтовое оформление, цветовая палитра. Также был разработан дизайн пользовательских элементов.

3. Реализация структуры веб-сайта

Представить листинги структуры веб-сайта. Обосновать выбор элементов в коде.

Если это большой важный кусок кода, то покажите его в приложение в виде листинга, но укажите ссылку на приложение. Например, скрипт для создания таблиц представлен в приложении Б. В листингах интервалов после абзацев быть не должно.

**3.1 Структура HTML-документа**

HTML-документ представляет собой основу веб-страницы, определяющую её структуру и содержимое.

При формировании хорошей структуры HTML-документа необходимо учитывать несколько важных аспектов:

1. Семантика: Использование семантических элементов HTML (например, <header>, <nav>, <article>, <footer>) помогает браузерам и поисковым системам правильно интерпретировать содержимое страницы, что улучшает доступность и SEO-оптимизацию.
2. Читабельность и понятность: Хорошо организованная структура делает код более читаемым для других разработчиков, что упрощает совместную работу и обслуживание проекта.
3. Масштабируемость и обслуживаемость: Правильная структура облегчает добавление нового содержимого или внесение изменений в будущем, так как она делает код более модульным и понятным.
4. Доступность: Грамотно построенная структура обеспечивает лучшую доступность для пользователей с ограниченными возможностями, так как она позволяет использовать вспомогательные технологии, такие как скринридеры.
5. Производительность: Эффективная структура помогает ускорить загрузку страницы, поскольку браузеры могут быстрее понять, как отобразить содержимое.

Структура главной страницы данного курсового проекта представлена в листинге 3.1.

**3.2 Добавление таблиц стилей SCSS и CSS**

В ходе разработки был применен внешний способ подключения таблиц стилей, что продемонстрировано в листинге 3.2.

<head>

...

   <link rel="stylesheet" href="css/1.css">

...

</head>

Листинг 3.2 – Пример внешнего подключения

Таблицы стилей были реализованы преимущественно с использованием селекторов по классу, так как они способствуют более четкой структуре кода, упрощают его понимание и обеспечивают более гибкую возможность стилизации элементов. Это позволяет создавать более модульный и масштабируемый код, что улучшает его поддержку и развитие в будущем.

В листинге 3.3 расположен пример таблицы стилей SCSS, а в листинге 3.4 расположен пример таблицы стилей CSS.

#header {

  display: flex;

  align-items: center;

  height: 100px;

  margin: 0 auto;

  width: 90%;

  max-width: 1425px;

  > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

  }

Листинг 3.3 – Пример таблицы стилей SCSS

SCSS (Sassy CSS) является предварительным процессором CSS, который добавляет мощные функции и улучшает организацию и поддерживаемость кода CSS. Можно выделить несколько преимуществ SCSS перед CSS:

1. Переменные: SCSS позволяет определять переменные для значений, таких как цвета, размеры шрифтов, отступы и т. д. Это упрощает повторное использование и обновление значений по всему коду.
2. Вложенные правила: SCSS позволяет вкладывать CSS-правила внутрь других правил, что делает код более читабельным и удобным для организации.
3. Миксины: Миксины в SCSS позволяют определять и повторно использовать группы CSS-свойств. Это особенно полезно для создания кроссбраузерных стилей, анимаций и других повторяющихся блоков кода.
4. Вложенные селекторы: SCSS позволяет использовать вложенные селекторы, что упрощает написание стилей для вложенных элементов без необходимости повторного указания родительских селекторов.

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: 20%;

    width: 1425px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

Листинг 3.4 – Пример таблицы стилей CSS

В данном примере мы устанавливаем стиль шрифта текста на «Montserrat» и белый цвет текста; задаем фоновое изображение страницы, которое занимает всю доступную область и не повторяется; определяем ширину заголовка на 100% экрана, выравниваем его по центру и добавляем отступ сверху в 52 пикселя; автоматически масштабируем изображение до 100% ширины родительского элемента.

## **3.3. Использование стандартов XML (SVG)**

Для создания иконки логотипа был выбран SVG-формат хранения. В листинге 3.5 продемонстрирован пример.

 <svg width="10"height="100"xmlns="http://www.w3.org/2000/svg"class="header\_rod">

 <rect width="2" height="80" fill="black"/>

 </svg>

Листинг 3.5 – Пример иконки в SVG-формате

В XML-формате удобно хранить данные, которые выражены в иерархической структуре и могут быть легко интерпретированы как человеком, так и компьютером. Это могут быть данные различных типов, такие как текстовые документы, конфигурационные файлы, данные о товарах и услугах, организационные схемы, информация о контактах, настройки приложений и многое другое. XML обеспечивает гибкость и расширяемость для представления таких данных, позволяя создавать собственные наборы тегов и определять структуру данных в соответствии с конкретными потребностями приложения или системы.

Пример хранения данных в XML-формате проиллюстрирован в листинге 3.6.

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="footer.xsl"?>

<authorInfo>

    <surname>Мысливцев</surname>

    <name>Алексей</name>

    <faculty>ФИТ</faculty>

    <spec>ИСиТ (проф. ИС) </spec>

    <course>1-го</course>

    <group>1-ой</group>

    <topic>Веб-сайт прогноза погоды</topic>

</authorInfo>

Листинг 3.6 – Пример хранения данных в XML-формате

Данный пример отвечает за хранение данных о студенте. Вышеперечисленные данные в будущем будут отображены на веб-сайте.

3.4 Использование JavaScript

JavaScript (JS) — это высокоуровневый, интерпретируемый язык программирования, который применяется в веб-разработке для создания интерактивных веб-страниц. JS позволяет добавлять динамическое поведение к веб-сайтам, взаимодействовать с пользователем, управлять содержимым страницы, обрабатывать события, валидировать данные и многое другое.

Пример использования JavaScipt проиллюстрирован в листинге 3.7.

 function formatDate(dateString) {

        const date = new Date(dateString);

        const options = { weekday: 'short', day: 'numeric', month: 'long' };

        let formattedDate = date.toLocaleDateString('ru-RU', options);

        // Преобразование первой буквы дня недели в заглавную

        formattedDate = formattedDate.charAt(0).toUpperCase() + formattedDate.slice(1);

        return formattedDate;

    }

Листинг 3.7 – Пример использования JavaScript для реализации функции преобразования даты

## **3.4. Выводы**

В данном разделе была реализована структура веб-сайта, а именно: реализована структура HTML-документа, добавлены таблицей стилей SCSS и CSS, было продемонстрировано использование стандартов XML, SVG, а также языка программирования JavaScript.

4. Тестирование веб-сайта

## **4.1. Адаптивный дизайн веб-сайта**

Представить как будет осуществляться адаптивность вашего сайта под разные устройста в различных браузерах

Для достижения адаптивности, помимо использования flexbox-контейнеров, использовались медиа-запросы.

Если сайт адаптивный, то он должен корректно отображаться на экранах различных устройств: планшетные устройства (рис 4.1), компьютер (рис 4.2).

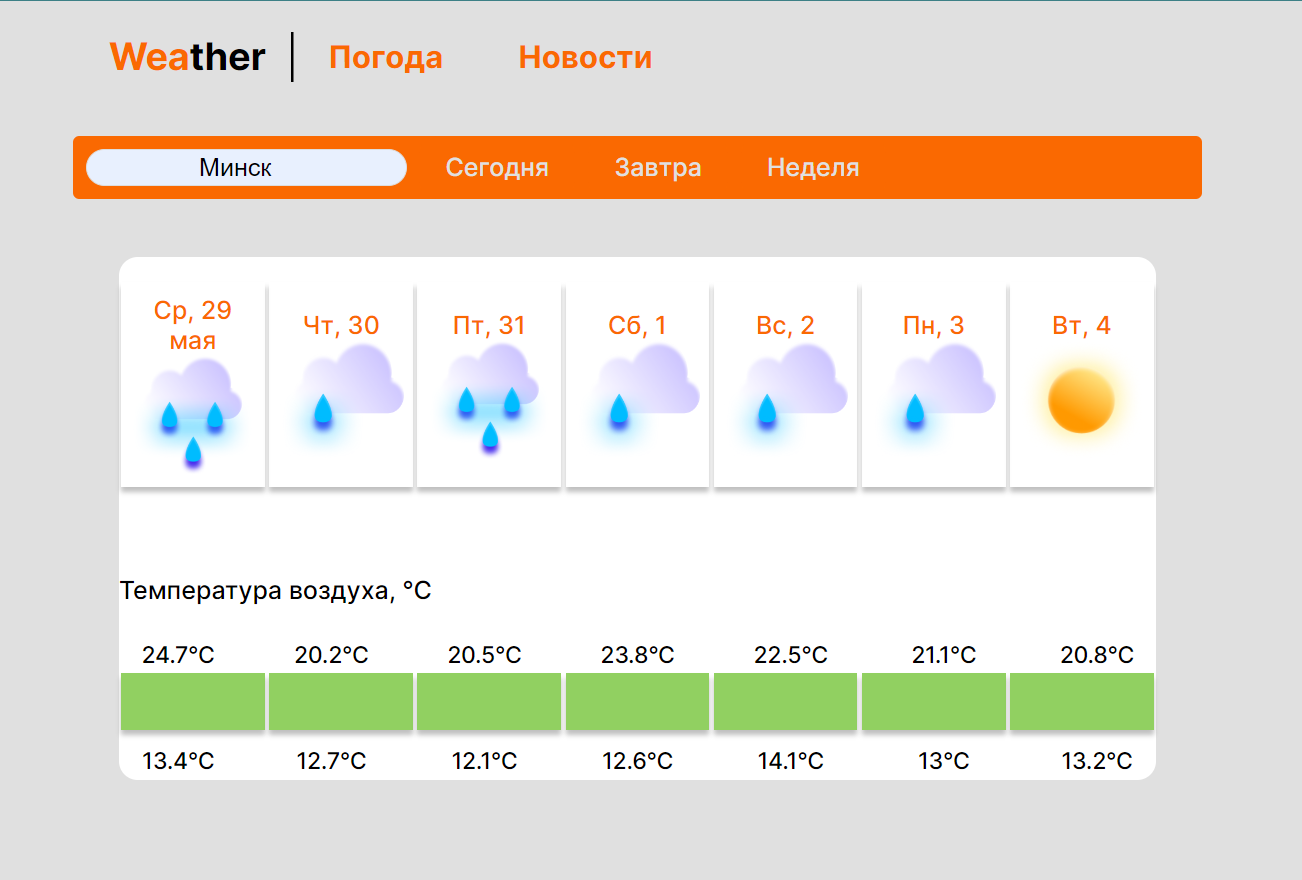


Рисунок 4.2 – Вид сайта с планшетных устройств

Данный пример показывает нам вид веб-сайта, который открыли на планшетном устройстве, либо ноутбуке.

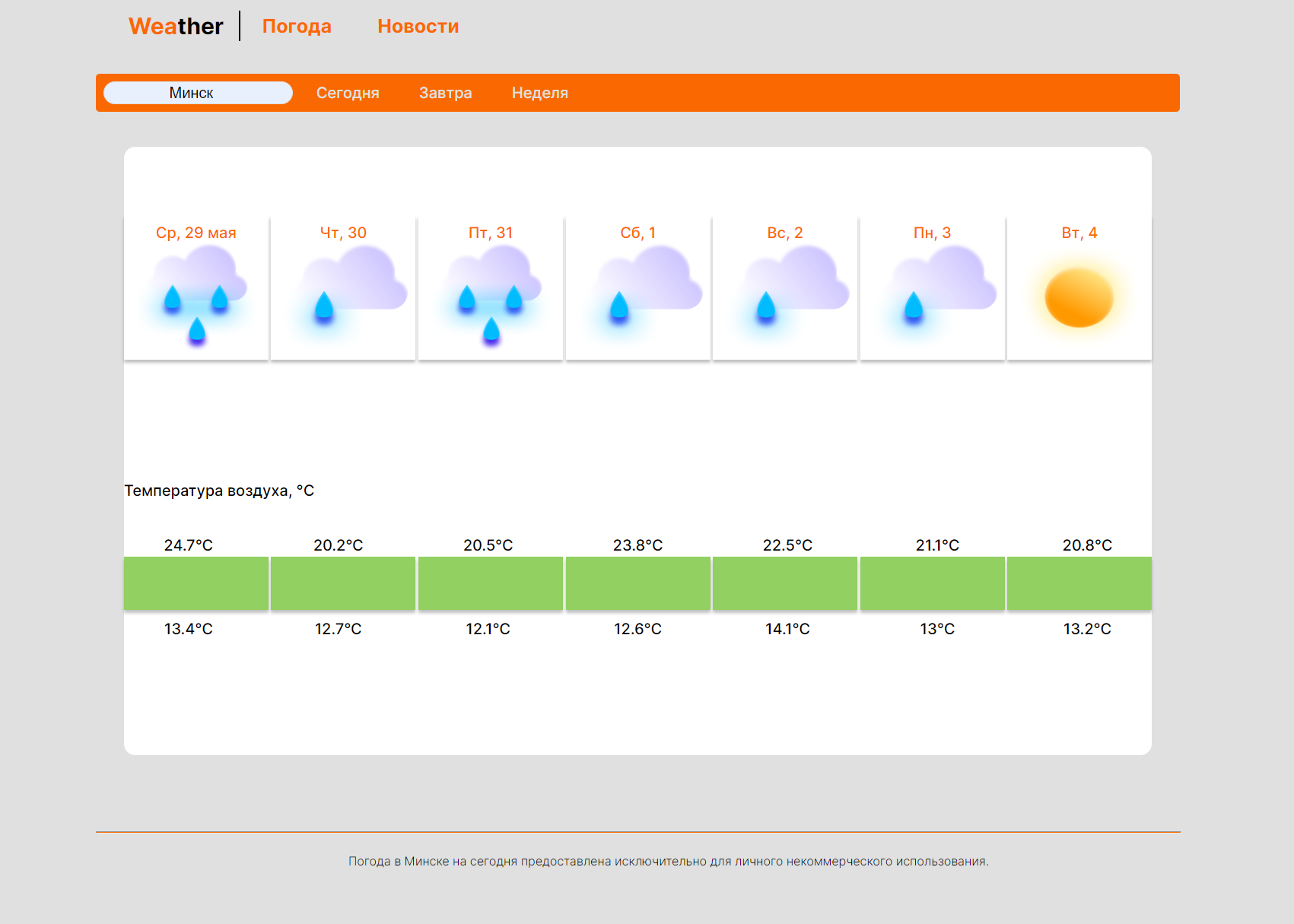


Рисунок 4.3 – Вид сайта с компьютера

Данный пример показывает нам вид веб-сайта, который открыли на компьютере.

Адаптивный дизайн веб-сайта позволяет ему корректно отображаться и работать на различных устройствах и экранах, таких как компьютеры, планшеты и ноутбуки. Вот несколько причин, почему адаптивность важна для сайта:

1. Пользовательский опыт (UX): Адаптивный дизайн обеспечивает удобство использования сайта для всех пользователей, независимо от того, какое устройство они используют. Это создает позитивное впечатление и повышает вероятность того, что пользователь останется на сайте.
2. SEO-оптимизация: Поисковые системы, такие как Google, предпочитают сайты с адаптивным дизайном, поскольку такие сайты обеспечивают более удобный доступ к контенту для пользователей, что положительно влияет на показатели SEO.
3. Доступность: Адаптивный дизайн делает сайт доступным для всех категорий пользователей, включая людей с ограниченными возможностями, так как он учитывает различные способы взаимодействия с сайтом.
4. Повышение конверсии: Хорошо спроектированный адаптивный сайт может увеличить конверсию, поскольку улучшает удобство использования и повышает вероятность выполнения целевых действий, таких как покупка товара или заполнение формы.
5. Экономия времени и ресурсов: Вместо создания отдельных версий сайта для разных устройств, адаптивный дизайн позволяет оптимизировать процесс разработки и обслуживания, что экономит время и ресурсы.

## **4.2. Кроссбраузерность веб-сайта**

В кроссбраузерное тестирование входит общий вид вашего проекта в других браузерах (сохранились ли шрифты, не съехала ли анимация и так далее)

Кроссбраузерность важна для обеспечения корректного отображения и работы вашего сайта на различных веб-браузерах. Это помогает создать одинаковый пользовательский опыт для всех посетителей и предотвращает потерю потенциальных клиентов из-за проблем с отображением или функциональностью на определенных браузерах, увеличивая тем самым охват аудитории и общее восприятие сайта.

На рисунке 4.4 продемонстрирован вид веб-сайта, открытого в браузере «Google Chrome», а на рисунке 4.5 продемонстрирован вид веб-сайта, открытого в браузере «Microsoft Edge».

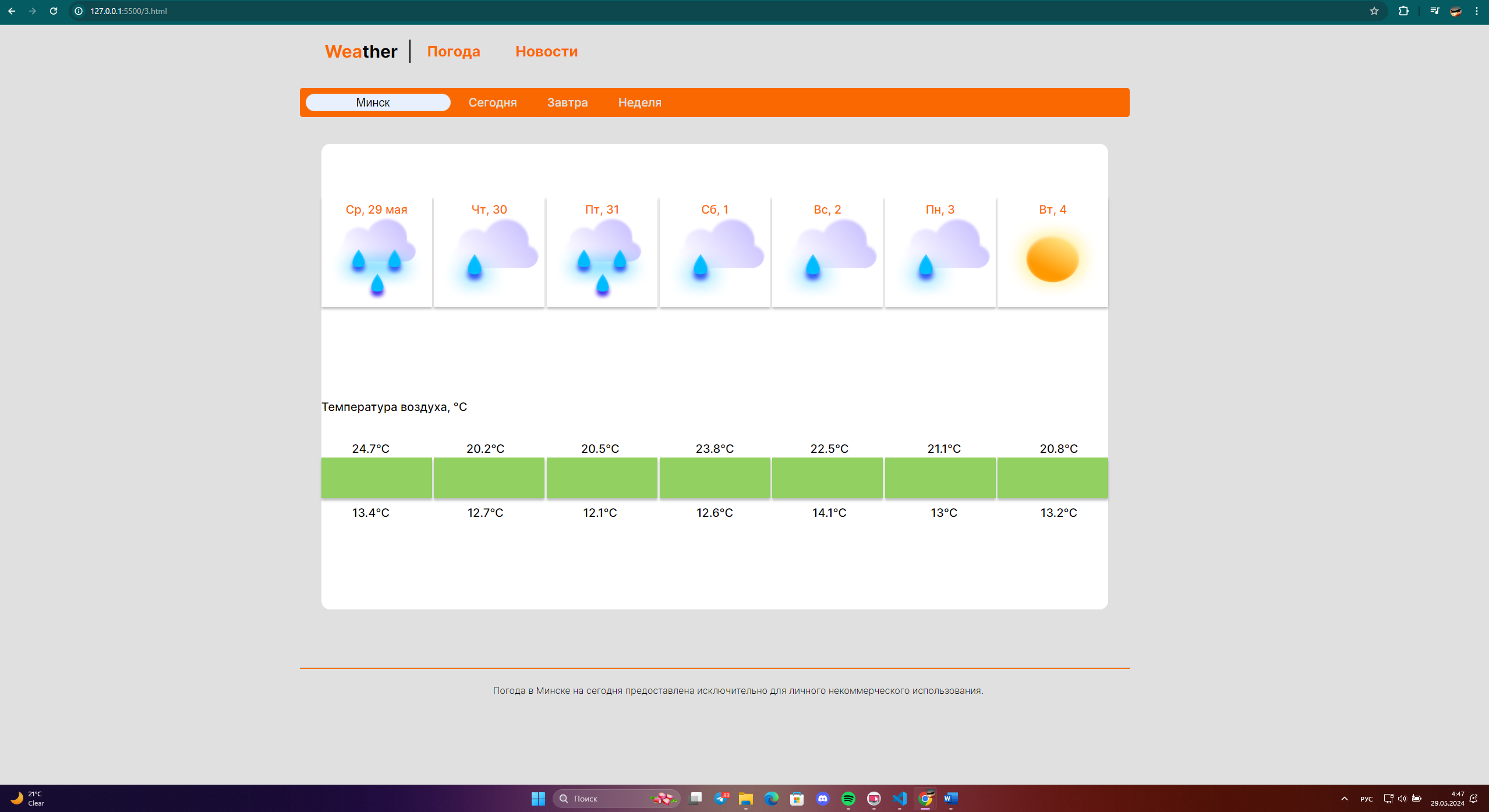


Рисунок 4.4 – Вид веб-сайта в браузере «Google Chrome»

На данном рисунке продемонстрирован вид веб-сайта, открытого в браузере «Google Chrome».

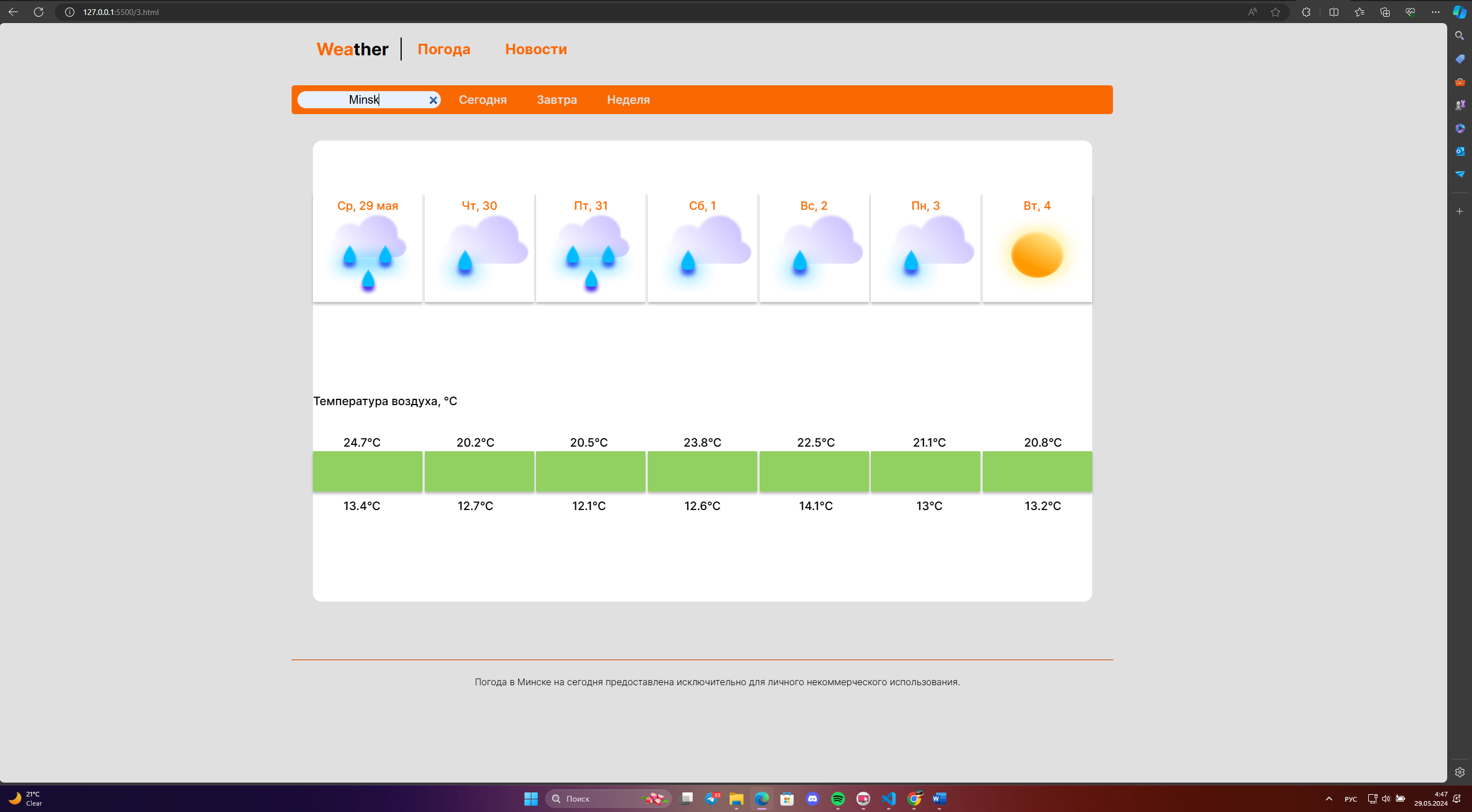


Рисунок 4.5 – Вид веб-сайта в браузере «Microsoft Edge»

На данном рисунке продемонстрирован вид веб-сайта, открытого в браузере «Microsoft Edge».

## **Тестирование кода**

В процессе разработки веб-сайта, тестирование кода играет критическую роль. Оно становится важным этапом, который позволяет тщательно проверить каждый аспект веб-проекта, начиная от его визуального представления и заканчивая функциональными возможностями.

Первое, что необходимо учитывать — это корректность отображения веб-страниц на различных устройствах и в разных браузерах. Каждый браузер может по-разному интерпретировать код, что может привести к непредвиденным отображениям и расхождениям в дизайне. Проведение тестирования поможет обнаружить такие проблемы и гарантировать, что сайт выглядит одинаково привлекательно и функционально на всех платформах.

Кроме того, тестирование кода необходимо для проверки скорости загрузки веб-страниц. Медленно загружающиеся страницы могут отпугнуть пользователей и ухудшить пользовательский опыт.

Тестирование на безопасность помогает выявить потенциальные уязвимости и угрозы, которые могут быть использованы злоумышленниками для атаки на ваш веб-ресурс.

Обнаружение и устранение уязвимостей на этапе тестирования позволяет снизить риск возможных атак и обеспечить защиту конфиденциальности и целостности данных вашего веб-сайта, что является крайне важным аспектом в современной интернет-среде, где угрозы безопасности постоянно эволюционируют.

Для тестирования кода был выбран сервис The W3C Markup Validation Service. Результат тестирования представлен на рисунке 4.8.

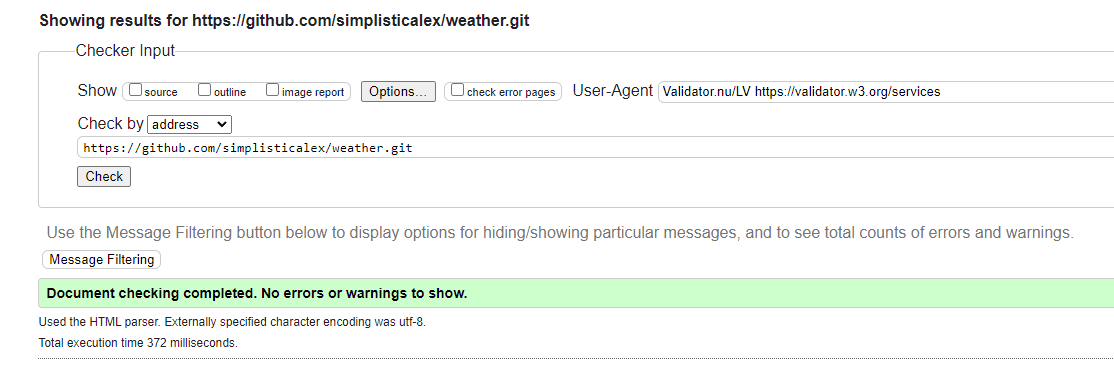


Рисунок 4.8 – результат тестирования

Опираясь на данный рисунок, можно сказать, что сайт был написан корректно.

4.5 Выводы

В данном разделе было рассмотрено поведение веб-сайта на различных устройствах и в различных браузерах. Также в данном разделе было проведено тестирование кода веб-сайта, были оговорены преимущества проведения тестирования.

Заключение

В ходе выполнения курсовой работы был реализован познавательный веб-сайт на тему «Солнечная система» для ознакомления пользователей с небесными телами космоса

При разработке данного веб-сайта был использован широкий ряд языков разметки и таблиц стиля, а именно: HTML5, SCSS, CSS3, JS, XML, которое позволило повысить качество продукта и уменьшить временные затраты на его разработку в несколько раз.

При постановке задачи было рассмотрено некоторые количество аналогичных решений. Были рассмотрены преимущества и недостатки того или иного сайта, в результате чего было составлено техническое задание.

При проектировании веб-сайта были разработаны прототипы, далее создан макет дизайна веб-сайт, опираясь на который происходила сама вёрстка.

Для разработки макета сайта была использована онлайн-программа «Figma».

В ходе тестирования веб-сайта была продемонстрирована его кроссбраузерность и адаптивность.

В пояснительной записке к данному курсовому проекту описана реализация поставленных в рамках данного проекта задач:

1. Постановка задачи. Анализ готовых решений и выбор средств реализации.
2. Проектирование веб-сайта (выбор способа верстки, выбор стилевого и шрифтового оформлений, и так далее)
3. Реализация структура веб-сайта посредством использования HTML, CSS, SCSS и JS.
4. Тестирование веб-сайта на кроссбраузерность, адаптивность.

В моей курсовой работе был представлен комплексный анализ и разработка веб-сайта прогноза погоды. Проект охватывает все ключевые процессы, начиная от исследования и выбора технологий, до детального разбора архитектуры, дизайна пользовательского интерфейса и вопросов совместимости с различными платформами.

В процессе разработки особое внимание было уделено обеспечению интуитивно понятного дизайна для получения актуальных погодных данных и использования современных подходов к визуализации.

Адаптивность приложения к различным платформам и устройствам подчеркивает его универсальность и перспективы для широкого распространения. Это не только увеличивает его потенциальную пользовательскую базу, но и открывает новые возможности для дальнейшего развития и расширения функционала.

Проект представляет собой ценный опыт в моем обучении и профессиональном развитии.

6. Список использованных литературных источников.

1. Айзекс, С. Dynamic HTML. Секреты создания интерактивных Web-страниц / С. Айзекс. – СПб.: БХВ-Петербург, 1998.

2. Гончаров, А. Самоучитель HTML / А. Гончаров. – СПб.:Питер, 2002.

3.Приложение для разработки макетов и прототипов веб-страниц «Figma» [Электронный ресурс]. – Режим доступа: https://www.figma.com/

4.Документация по HTML/CSS/JavaScript [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/

5.Справочник по HTML [Электронный ресурс]. – Режим доступа: https://htmlbook.ru/html

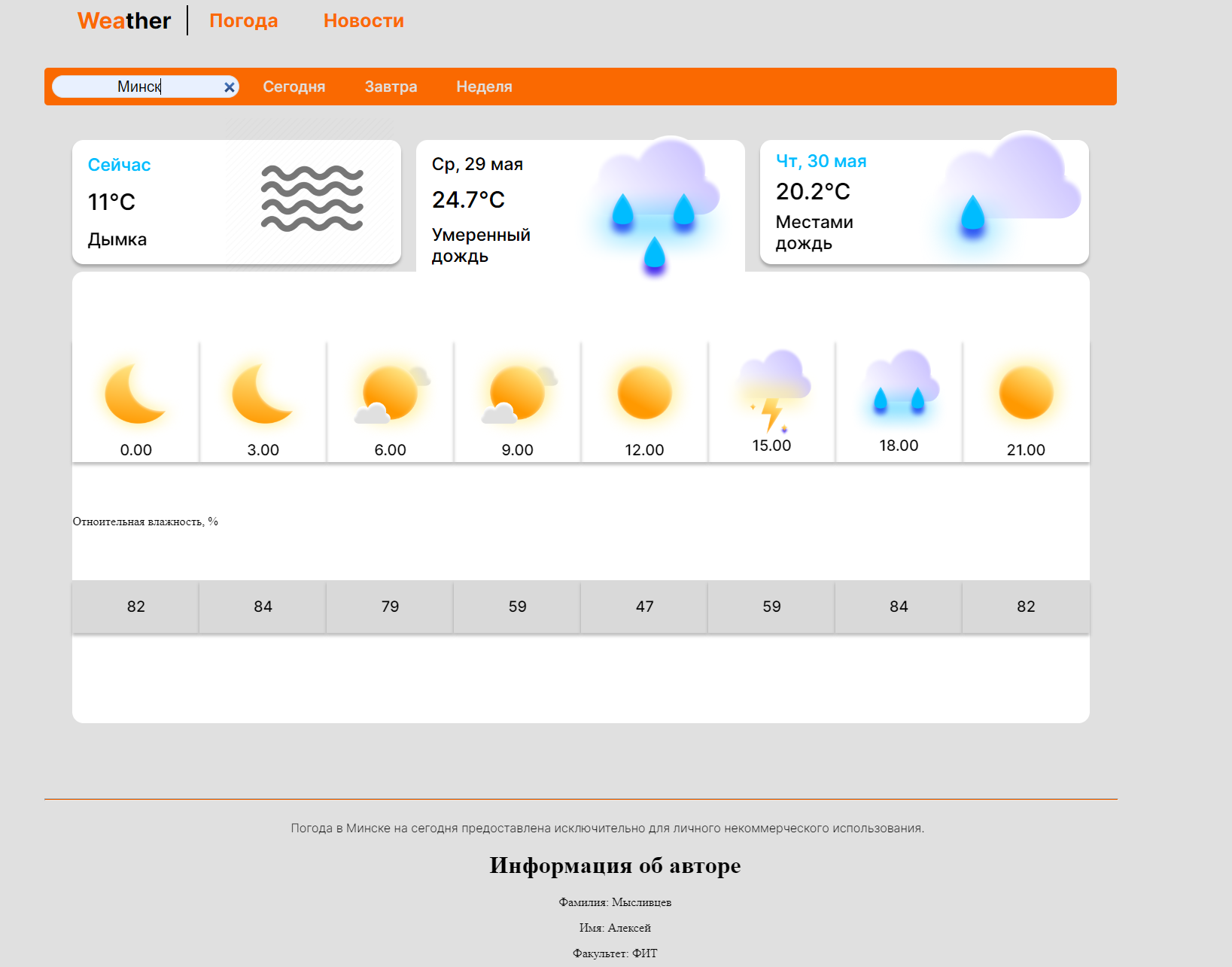
6.Документация по SCSS [Электронный ресурс]. – Режим доступа: https://sass-scss.ru/

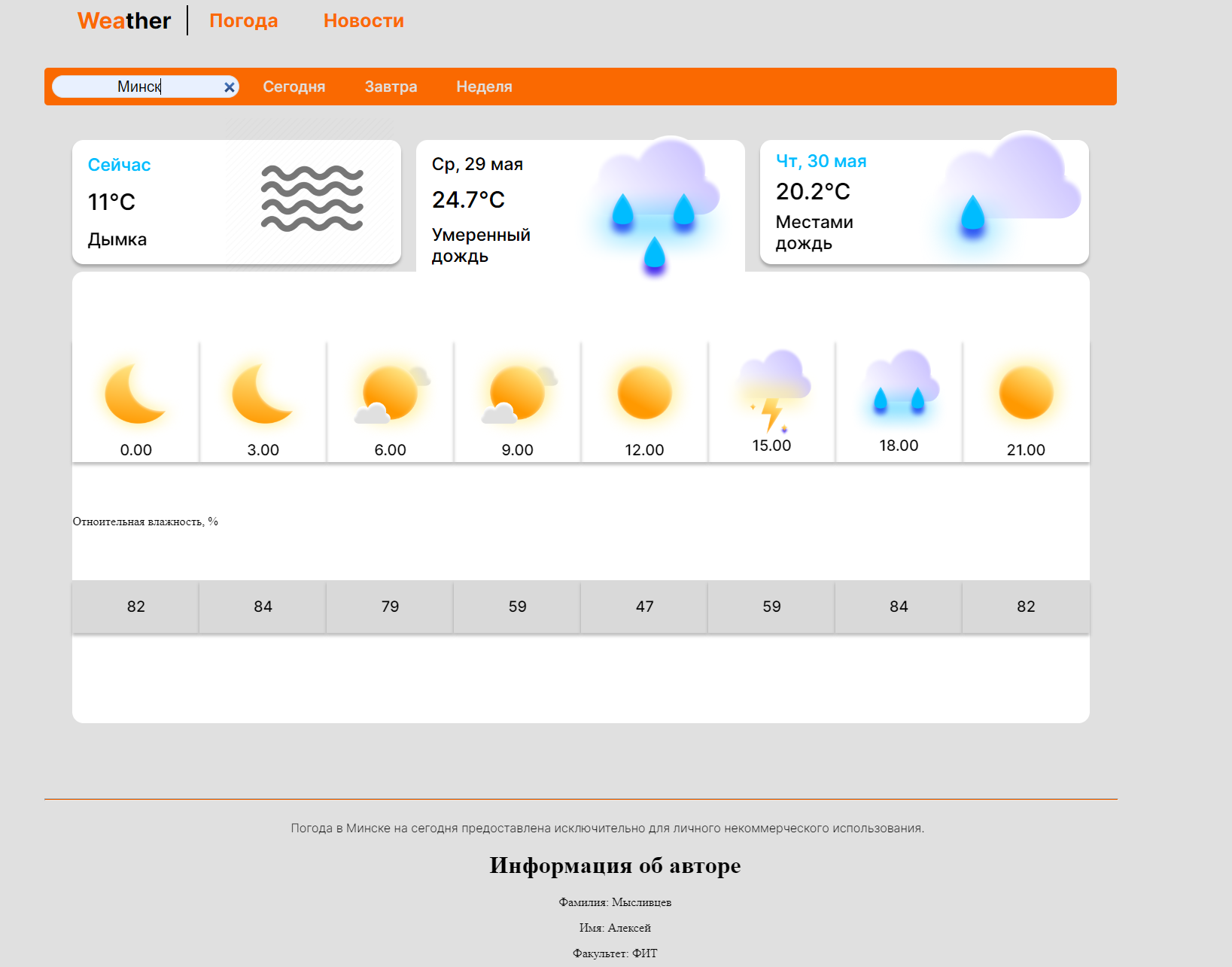
7.Репозиторий проекта на GitHub [Электронный ресурс]. — Режим доступа: https://github.com/simplisticalex/weather

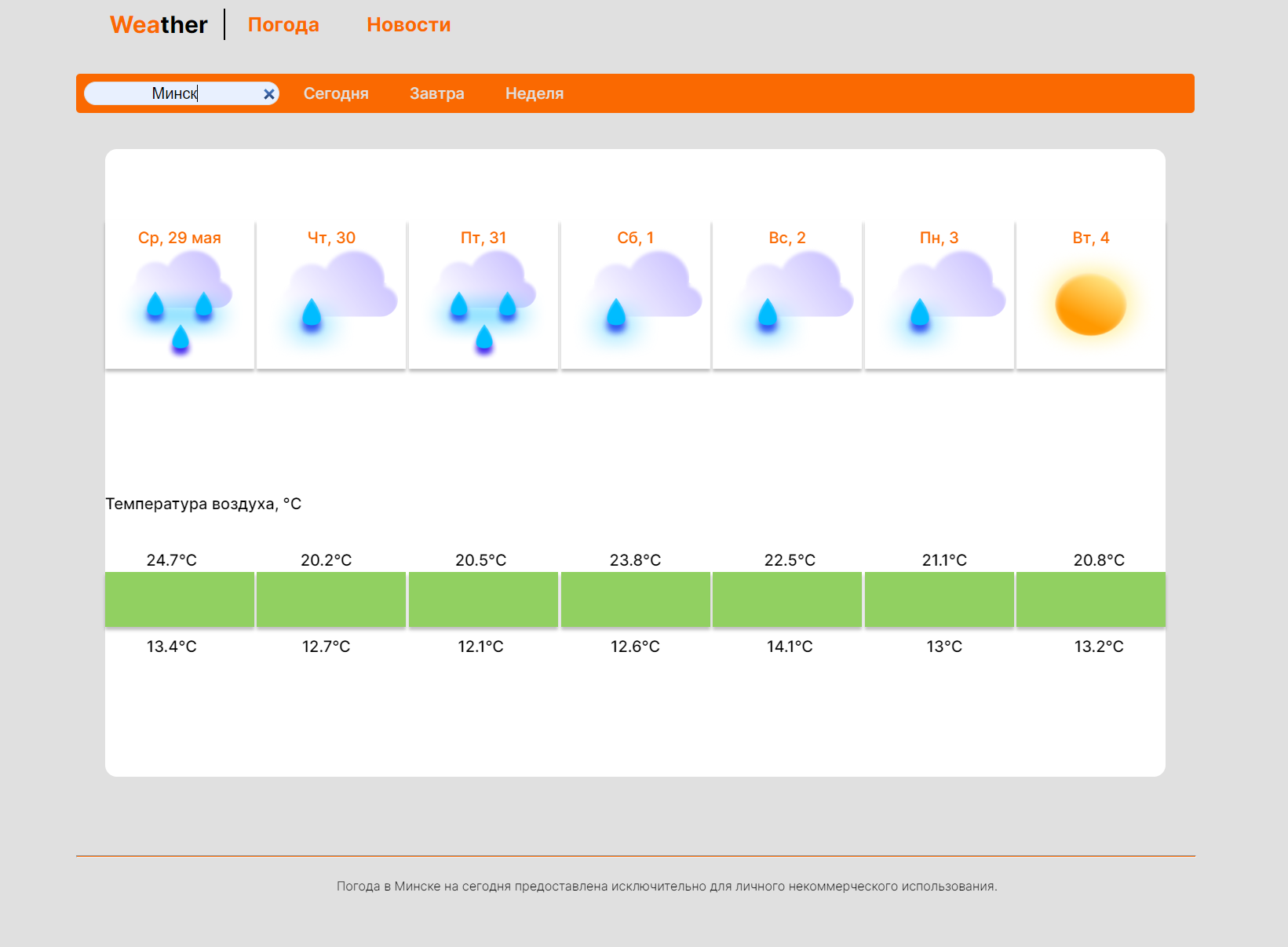
8.https://www.mvideo.ru/blog/podborki/10-samyh-tochnyh-servisov-prognoza-pogody

7. Приложение (полный исходный текст программы разработанного приложения с подробными комментариями).

ПРИЛОЖЕНИЕ А. Макет структуры веб-сайта







ПРИЛОЖЕНИЕ Б. Листинг HTML-документа

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="css/1.css">

    <script>

        //Загрузка XML файлов

        function loadXMLDoc(filename) {

            return new Promise((resolve, reject) => {

                const xhttp = new XMLHttpRequest();

                xhttp.onreadystatechange = function() {

                    if (this.readyState == 4 && this.status == 200) {

                        resolve(this.responseXML);

                    } else if (this.readyState == 4 && this.status != 200) {

                        reject(new Error(`Failed to load ${filename}`));

                    }

                };

                xhttp.open("GET", filename, true);

                xhttp.send();

            });

        }

        //примен xsl к xml, вывод результ

        async function displayResult() {

            try {

                const xml = await loadXMLDoc("1.xml");

                const xsl = await loadXMLDoc("footer.xsl");

                if (window.ActiveXObject || "ActiveXObject" in window) { // IE

                    const ex = xml.transformNode(xsl);

                    document.getElementById("example").innerHTML = ex;

                } else if (document.implementation && document.implementation.createDocument) { //Chrome, Firefox, Opera, etc.

                    const xsltProcessor = new XSLTProcessor();

                    xsltProcessor.importStylesheet(xsl);

                    const resultDocument = xsltProcessor.transformToFragment(xml, document);

                    document.getElementById("example").appendChild(resultDocument);

                }

            } catch (error) {

                console.error('Error:', error);

            }

        }

        document.addEventListener("DOMContentLoaded", displayResult);

    </script>

</head>

<body>

    <header>

        <div id="header">

            <h1 id="Weather">Wea<span id="black\_weather">ther</span></h1>

            <svg width="10" height="100" xmlns="http://www.w3.org/2000/svg" class="header\_rod">

                <rect width="2" height="80" fill="black"/>

            </svg>

            <h2 id="pogoda"><a href="1.html">Погода</a></h2>

            <h2 id="novosti"><a href="https://news.google.com/home?hl=ru&gl=RU&ceid=RU:ru">Новости</a></h2>

        </div>

    </header>

    <nav>

        <div id="navbar">

            <form action="/login" id="form" method="post">

                <input type="search" id="input" placeholder="Город">

            </form>

            <button><a href="1.html" id="today">Сегодня</a></button>

            <button><a href="2.html" id="tomorrow">Завтра</a></button>

            <button><a href="3.html" id="Week">Неделя</a></button>

        </div>

    </nav>

    <br>

    <br id="br">

    <!-- <article id="article">

        /\*<div id="flex-container">

            <div id="perviy">

                <div class="card">

                    <div class="card-time">Сейчас</div>

                    <div class="card-temp">14°C</div>

                    <div class="card-weather">Ясно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="vtoroy">

                <div class="card">

                    <div class="card-time2">Пт, 5 апреля</div>

                    <div class="card-temp">15°C</div>

                    <div class="card-weather">Облачно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="tretiy">

                <div class="card">

                    <div class="card-time">Сб, 6 апреля</div>

                    <div class="card-temp">8°C</div>

                    <div class="card-weather">Дождь</div>

                </div>

                <img id="weather-icon" src="pictures/sunny.png" alt="sunny">

            </div>

        </div>

        <div id="flex-container2">

            <div id="chetv">

                <div class="row top-row">

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 0.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 3.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 6.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 9.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 12.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 15.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 18.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 21.00</div>

                </div>

                <div id="humid">Отноительная влажность, %</div>

                <div class="row bottom-row">

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >84</div>

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >61</div>

                    <div class="chetv-inner" >45</div>

                    <div class="chetv-inner" >38</div>

                    <div class="chetv-inner" >35</div>

                    <div class="chetv-inner" >51</div>

                </div>

            </div>

        </div>

    </article> -->

    <footer>

        <hr>

        <br>

        <div id="last">Погода в Минске на сегодня предоставлена исключительно для личного некоммерческого использования.</div>

        <div id="last"><xml src="1.xml"></xml></div>

        <div id="example"></div>

    </footer>

    <script src="main.js" type="module"></script>

    <script src="conditions.js" type="module"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="2.css">

</head>

<body>

    <header>

        <div id="header">

            <h1 id="Weather">Wea<span id="black\_weather">ther</span></h1>

            <svg width="10" height="100" xmlns="http://www.w3.org/2000/svg" class="header\_rod">

                <rect width="2" height="80" fill="black"/>

            </svg>

            <h2 id="pogoda"><a href="1.html">Погода</a></h2>

            <h2 id="novosti"><a href="https://news.google.com/home?hl=ru&gl=RU&ceid=RU:ru">Новости</a></h2>

        </div>

    </header>

    <nav>

        <div id="navbar">

            <form action="/login" id="form" method="post">

            <input type="search" id="input" placeholder="Город"></form>

            <button><a href="1.html" id="today">Сегодня</a></button>

            <button><a href="2.html" id="tomorrow">Завтра</a></button>

            <button><a href="3.html" id="Week">Неделя</a></button>

        </div>

    </nav>

    <br>

    <br id="br">

    <!-- <article id="article">

        /\*<div id="flex-container">

            <div id="perviy">

                <div class="card">

                    <div class="card-time">Сейчас</div>

                    <div class="card-temp">14°C</div>

                    <div class="card-weather">Ясно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="vtoroy">

                <div class="card">

                    <div class="card-time2">Пт, 5 апреля</div>

                    <div class="card-temp">15°C</div>

                    <div class="card-weather">Облачно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="tretiy">

                <div class="card">

                    <div class="card-time">Сб, 6 апреля</div>

                    <div class="card-temp">8°C</div>

                    <div class="card-weather">Дождь</div>

                </div>

                <img id="weather-icon" src="pictures/sunny.png" alt="sunny">

            </div>

        </div>

        <div id="flex-container2">

            <div id="chetv">

                <div class="row top-row">

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 0.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 3.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 6.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 9.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 12.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 15.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 18.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 21.00</div>

                </div>

                <div id="humid">Отноительная влажность, %</div>

                <div class="row bottom-row">

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >84</div>

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >61</div>

                    <div class="chetv-inner" >45</div>

                    <div class="chetv-inner" >38</div>

                    <div class="chetv-inner" >35</div>

                    <div class="chetv-inner" >51</div>

                </div>

            </div>

        </div>

    </article> -->

    <footer>

        <hr>

        <br>

        <div id="last">Погода в Минске на сегодня предоставлена исключительно для личного некоммерческого использования.</div>

    </footer>

    <script src="second.js" type="module"></script>

    <script src="conditions.js" type="module"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="3.css">

</head>

<body>

    <header>

        <div id="header">

            <h1 id="Weather">Wea<span id="black\_weather">ther</span></h1>

            <svg width="10" height="100" xmlns="http://www.w3.org/2000/svg" class="header\_rod">

                <rect width="2" height="80" fill="black"/>

            </svg>

            <h2 id="pogoda"><a href="1.html">Погода</a></h2>

            <h2 id="novosti"><a href="https://news.google.com/home?hl=ru&gl=RU&ceid=RU:ru">Новости</a></h2>

        </div>

    </header>

    <nav>

        <div id="navbar">

            <form action="/login" id="form" method="post">

            <input type="search" id="input" placeholder="Город"></form>

            <button><a href="1.html" id="today">Сегодня</a></button>

            <button><a href="2.html" id="tomorrow">Завтра</a></button>

            <button><a href="3.html" id="Week">Неделя</a></button>

        </div>

    </nav>

    <br>

    <br id="br">

    <!-- <article id="article">

        /\*<div id="flex-container">

            <div id="perviy">

                <div class="card">

                    <div class="card-time">Сейчас</div>

                    <div class="card-temp">14°C</div>

                    <div class="card-weather">Ясно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="vtoroy">

                <div class="card">

                    <div class="card-time2">Пт, 5 апреля</div>

                    <div class="card-temp">15°C</div>

                    <div class="card-weather">Облачно</div>

                </div>

                <img id="weather-icon" src="pictures/day/sunny.png" alt="sunny">

            </div>

            <div id="tretiy">

                <div class="card">

                    <div class="card-time">Сб, 6 апреля</div>

                    <div class="card-temp">8°C</div>

                    <div class="card-weather">Дождь</div>

                </div>

                <img id="weather-icon" src="pictures/sunny.png" alt="sunny">

            </div>

        </div>

        <div id="flex-container2">

            <div id="chetv">

                <div class="row top-row">

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 0.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 3.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 6.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 9.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 12.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 15.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 18.00</div>

                    <div class="chetv-inner"><img src="pictures/sunny.png" alt="sunny"> 21.00</div>

                </div>

                <div id="humid">Отноительная влажность, %</div>

                <div class="row bottom-row">

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >84</div>

                    <div class="chetv-inner" >79</div>

                    <div class="chetv-inner" >61</div>

                    <div class="chetv-inner" >45</div>

                    <div class="chetv-inner" >38</div>

                    <div class="chetv-inner" >35</div>

                    <div class="chetv-inner" >51</div>

                </div>

            </div>

        </div>

    </article> -->

    <footer>

        <hr>

        <br>

        <div id="last">Погода в Минске на сегодня предоставлена исключительно для личного некоммерческого использования.</div>

    </footer>

    <script src="third.js" type="module"></script>

    <script src="conditions.js" type="module"></script>

</body>

</html>

ПРИЛОЖЕНИЕ В. Листинг SCSS

@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&family=Inter:wght@100..900&family=Pixelify+Sans:wght@700&display=swap');

// Основные стили

body {

  background-color: #e0e0e0;

  margin: 0;

  padding: 0;

}

#header {

  display: flex;

  align-items: center;

  height: 100px;

  margin: 0 auto;

  width: 90%;

  max-width: 1425px;

  > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

  }

  a {

    text-decoration: none;

    color: #fa6901;

    &:hover {

      color: #e06004;

    }

  }

  h1 {

    margin-left: 3%;

    font-size: 30px;

  }

  #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

  }

  .header\_rod {

    height: 40px;

  }

  #black\_weather {

    color: #000;

  }

}

#navbar {

  background-color: #fa6901;

  border: 1px solid;

  display: flex;

  justify-content: flex-start;

  align-items: center;

  height: 50px;

  padding: 0 10px;

  margin: 0 auto;

  width: 90%;

  max-width: 1405px;

  border: none;

  border-radius: 5px;

  a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

  }

  input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

    &::placeholder {

      color: #fa6901;

    }

  }

  button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

    &:hover {

      background-color: #e06004;

    }

  }

}

article {

  margin: 0 auto;

  width: 90%;

  max-width: 1425px;

}

#perviy, #vtoroy, #tretiy {

  height: 165px;

  width: 437px;

  display: flex;

  align-items: center;

  justify-content: space-between;

  padding: 10px;

  box-sizing: border-box;

  background-color: #ffff;

}

#perviy, #tretiy {

  box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

  border-radius: 15px;

}

#perviy {

  margin-right: 20px;

}

#vtoroy {

  border-bottom: none;

  height: 185px;

  position: relative;

  top: 10px;

  border-radius: 15px 15px 0 0;

}

#tretiy {

  margin-left: 20px;

}

#flex-container, #flex-container2 {

  display: flex;

  margin: 0 auto;

  align-items: center;

  width: 90%;

  max-width: 1425px;

}

#chetv {

  width: 90%;

  max-width: 1351.7px;

  display: flex;

  flex-direction: column;

  justify-content: space-between;

  height: 600px;

  border-top: none;

  background-color: #ffff;

  border-radius: 15px;

  margin: 0 auto;

}

.chetv-inner {

  width: 150px;

  display: flex;

  align-items: center;

  justify-content: center;

  flex-direction: column;

  text-align: center;

  box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

  padding: 10px;

  margin-top: 90px;

  font-family: Inter;

  font-weight: 500;

  font-size: 20px;

  height: 143px;

  img {

    width: 136px;

    height: 130px;

  }

}

#chetv .row {

  display: flex;

  justify-content: space-between;

}

.bottom-row .chetv-inner {

  margin-bottom: 120px;

  width: 166px;

  height: 50px;

  margin-top: 0;

  background-color: #d9d9d9;

}

#combined {

  display: flex;

  flex-direction: column;

  width: 28%;

}

.card {

  display: flex;

  flex-direction: column;

  align-items: flex-start;

  justify-content: space-around;

  width: 100%;

  height: 100%;

  font-size: 23px;

  font-weight: 550;

  font-family: Inter;

  padding-left: 10px;

  > div {

    width: 100%;

    display: flex;

    justify-content: space-between;

  }

  .card-temp {

    font-size: 30px;

  }

  .card-time {

    color: #00bcff;

  }

  #weather-icon {

    max-width: 100%;

    height: auto;

  }

  #humid {

    margin-top: 20px; // Добавлено для отступа сверху

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    padding-bottom: 10px;

  }

}

hr {

  height: 1px;

  margin: 100px auto 0;

  width: 90%;

  max-width: 1425px;

  background-color: #fa6901;

}

#last {

  margin-right: 20px;

  text-align: center;

  font-family: Inter;

  font-size: 16px;

  font-weight: 300;

}

// Средние экраны (планшеты)

@media (min-width: 768px) and (max-width: 1024px) {

  body {

    background-color: #e0e0e0;

  }

  #header {

    flex-direction: column;

    align-items: center;

    margin: 0;

    width: 100%;

  }

  #navbar {

    flex-direction: column;

    align-items: center;

    margin: 0;

    width: 100%;

  }

  article {

    margin: 0;

    width: 100%;

  }

  #perviy, #vtoroy, #tretiy {

    height: auto;

    width: 100%;

    margin: 10px 0;

  }

  #flex-container, #flex-container2 {

    flex-direction: column;

    align-items: center;

  }

  #chetv {

    width: 100%;

    height: auto;

  }

  .chetv-inner {

    width: 80px;

    margin-top: 40px;

    height: 120px;

    img {

      width: 100px;

      height: 96px;

    }

  }

  .bottom-row .chetv-inner {

    height: 30px;

  }

  .card {

    font-size: 18px;

    .card-temp {

      font-size: 20px;

    }

    #humid {

      margin-top: 20px;

      font-size: 20px;

    }

  }

  hr {

    margin-top: 20px;

    width: 100%;

  }

  #last {

    margin-top: -20px;

  }

}

// Маленькие экраны (мобильные устройства)

@media (max-width: 767px) {

  body {

    background-color: #e0e0e0;

  }

  #header {

    flex-direction: column;

    align-items: center;

    margin: 0;

    width: 100%;

    > \* {

      margin: 10px 0;

      font-size: 20px;

    }

    h1 {

      margin-left: 0;

    }

    #pogoda, #karta, #novosti {

      margin-right: 0;

      font-size: 18px;

    }

  }

  #navbar {

    flex-direction: column;

    align-items: flex-start;

    margin: 0;

    width: 100%;

    a {

      margin: 10px 0;

    }

    input {

      width: 100%;

      margin-top: 10px;

    }

  }

  article {

    margin: 0;

    padding: 10px;

    width: 100%;

  }

  #perviy, #vtoroy, #tretiy {

    height: auto;

    width: 100%;

    margin: 10px 0;

    padding: 10px;

  }

  #flex-container, #flex-container2 {

    flex-direction: column;

    align-items: stretch;

    margin: 10px 0;

  }

  #chetv {

    width: 100%;

    height: auto;

    margin: 10px 0;

  }

  .chetv-inner {

    width: 100%;

    margin: 10px 0;

    height: auto;

    img {

      width: 100%;

      height: auto;

    }

  }

  .bottom-row .chetv-inner {

    margin-bottom: 10px;

    width: 100%;

    height: auto;

  }

  .card {

    font-size: 16px;

    .card-temp {

      font-size: 18px;

    }

    #humid {

      margin-top: 10px;

      font-size: 18px;

    }

  }

  hr {

    margin: 10px 0;

    width: 100%;

  }

  #last {

    margin-top: 10px;

    font-size: 14px;

  }

}

footer {

  text-align: center;

}

ПРИЛОЖЕНИЕ B. Листинг CSS

@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&family=Inter:wght@100..900&family=Pixelify+Sans:wght@700&display=swap');

@media (min-width: 1921px) {

body {

    background-color: #e0e0e0;

}

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: 20%;

    width: 1425px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

#header a {

    text-decoration: none;

    color: #fa6901;

}

#header a:hover {

    color: #e06004;

}

#header h1 {

    margin-left: 3%;

    font-size: 30px;

}

.header\_rod {

    height: 40px;

}

#header #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

}

#black\_weather {

    color: #000;

}

#navbar {

    background-color: #fa6901;

    border: 1px solid;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    height: 50px;

    padding: 0 10px;

    margin-left: 20%;

    width: 1405px;

    border: none;

    border-radius: 5px;

}

#navbar a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

}

#navbar input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

}

#navbar button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

}

#navbar button:hover {

    background-color: #e06004;

}

#navbar input::placeholder {

    color: #fa6901;

}

article {

    margin-left: 20%;

}

#perviy, #vtoroy, #tretiy {

    height: 165px;

    width: 437px;

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 10px;

    box-sizing: border-box;

    background-color: #FFFF;

}

#perviy {

    margin-right: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#vtoroy {

    border-bottom: none;

    height: 185px;

    position: relative;

    top: 10px;

    border-radius: 15px 15px 0px 0px;

}

#tretiy {

    margin-left: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#flex-container {

    display: flex;

    margin-left: 37px;

    align-items: center;

}

#flex-container > div {

    display: flex;

    justify-content: center;

    align-items: center;

}

#flex-container2 {

    margin-top: 10px;

    display: flex;

    margin-left: 37px;

}

#chetv {

    width: 1351.7px;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    height: 600px;

    border-top: none;

    background-color: #FFFF;

    border-radius: 15px;

}

.chetv-inner {

    width: 150px;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    text-align: center;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    padding: 10px;

    margin-top: 90px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    height: 143px;

}

#chetv .row {

    display: flex;

    justify-content: space-between;

}

.chetv-inner img {

    width: 136px;

    height: 130px;

}

.bottom-row .chetv-inner {

    margin-bottom: 120px;

    width: 166px;

    height: 50px;

    margin-top: 0px;

    background-color: #d9d9d9;

}

#combined {

    display: flex;

    flex-direction: column;

    width: 28%;

}

.card {

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: space-around;

    width: 100%;

    height: 100%;

    font-size: 23px;

    font-weight: 550;

    font-family: Inter;

    padding-left: 10px;

}

.card > div {

    width: 100%;

    display: flex;

    justify-content: space-between;

}

.card-temp {

    font-size: 30px;

}

.card-time {

    color: #00bcff;

}

#weather-icon {

    max-width: 100%;

    height: auto;

}

#humid {

    padding-top: 140px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    padding-bottom: 10px;

}

hr {

    height: 1px;

    margin-top: 100px;

    margin-left: 20%;

    width: 1425px;

    background-color: #fa6901;

}

#last {

    margin-right: 20px;

    text-align: center;

    font-family: Inter;

    font-size: 16px;

    font-weight: 300;

}

}

@media (min-width: 768px) and (max-width: 1920px) {

body {

    background-color: #e0e0e0;

}

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: auto;

    margin-right: auto;

    width: 900px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

#header a {

    text-decoration: none;

    color: #fa6901;

}

#header a:hover {

    color: #e06004;

}

#header h1 {

    margin-left: 3%;

    font-size: 30px;

}

.header\_rod {

    height: 40px;

}

#header #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

}

#black\_weather {

    color: #000;

}

#navbar {

    background-color: #fa6901;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    height: 50px;

    padding: 0 10px;

    margin-left: auto;

    margin-right: auto;

    width: 883px;

    border: none;

    border-radius: 5px;

}

#navbar a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

}

#navbar input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

}

#navbar button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

}

#navbar button:hover {

    background-color: #e06004;

}

#navbar input::placeholder {

    color: #fa6901;

}

article {

    margin-left: auto;

    margin-right: auto;

    width: 900px;

}

#perviy, #vtoroy, #tretiy {

    height: 100px;

    width: 264px;

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 10px;

    box-sizing: border-box;

    background-color: #FFFF;

    overflow: hidden;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#perviy {

    margin-right: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#vtoroy {

    border-bottom: none;

    height: 120px;

    position: relative;

    top: 10px; /\* Поднятие вверх \*/

    border-radius: 15px 15px 0px 0px;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#tretiy {

    margin-left: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#perviy img, #vtoroy img, #tretiy img {

    max-width: 100%;

    max-height: 100%;

    display: block;

}

#perviy .text, #vtoroy .text, #tretiy .text {

    overflow: hidden;

    text-overflow: ellipsis;

    white-space: nowrap;

}

#flex-container {

    display: flex;

    justify-content: center;

    align-items: center;

    margin: 20px 0;

}

#flex-container > div {

    display: flex;

    justify-content: center;

    align-items: center;

}

#flex-container2 {

    display: flex;

    justify-content: center;

    align-items: center;

    margin: 20px 0;

}

#chetv {

    width: 830px;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    height: 340px;

    border-top: none;

    background-color: #FFFF;

    border-radius: 15px;

    margin-left: auto;

    margin-right: auto;

    margin-top: -20px; /\* Поднятие вверх \*/

}

.chetv-inner {

    width: 80px;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    text-align: center;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    padding: 10px;

    margin-top: 40px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    height: 120px;

}

#chetv .row {

    display: flex;

    justify-content: space-between;

}

.chetv-inner img {

    width: 100px;

    height: 96px;

}

.bottom-row .chetv-inner {

    margin-bottom: 120px;

    width: 166px;

    height: 30px;

    margin-top: 0px;

    background-color: #d9d9d9;

}

#combined {

    display: flex;

    flex-direction: column;

    width: 28%;

}

.card {

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: space-around;

    width: 100%;

    height: 100%;

    font-size: 18px;

    font-weight: 550;

    font-family: Inter;

    padding-left: 10px;

}

.card > div {

    width: 100%;

    display: flex;

    justify-content: space-between;

}

.card-temp {

    font-size: 20px;

}

.card-time {

    color: #00bcff;

}

#weather-icon {

    max-width: 100%;

    height: auto;

}

#humid {

    padding-top: 20px;

    font-size: 20px;

    font-family: Inter;

    font-weight: 500;

}

hr {

    height: 1px;

    margin-top: 20px; /\* Уменьшение отступа сверху \*/

    margin-left: auto;

    margin-right: auto;

    width: 900px;

    background-color: #fa6901;

}

#last {

    margin-right: 20px;

    text-align: center;

    font-family: Inter;

    font-size: 16px;

    font-weight: 300;

    margin-top: -20px; /\* Поднятие вверх \*/

}

}

footer {

    text-align: center;

}

@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&family=Inter:wght@100..900&family=Pixelify+Sans:wght@700&display=swap');

@media (min-width: 1921px) {

body {

    background-color: #e0e0e0;

}

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: 20%;

    width: 1425px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

#header a {

    text-decoration: none;

    color: #fa6901;

}

#header a:hover {

    color: #e06004;

}

#header h1 {

    margin-left: 3%;

    font-size: 30px;

}

.header\_rod {

    height: 40px;

}

#header #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

}

#black\_weather {

    color: #000;

}

#navbar {

    background-color: #fa6901;

    border: 1px solid;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    height: 50px;

    padding: 0 10px;

    margin-left: 20%;

    width: 1405px;

    border: none;

    border-radius: 5px;

}

#navbar a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

}

#navbar input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

}

#navbar button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

}

#navbar button:hover {

    background-color: #e06004;

}

#navbar input::placeholder {

    color: #fa6901;

}

article {

    margin-left: 20%;

}

#perviy, #vtoroy, #tretiy {

    height: 165px;

    width: 437px;

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 10px;

    box-sizing: border-box;

    background-color: #FFFF;

}

#perviy {

    margin-right: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#vtoroy {

    border-bottom: none;

    height: 185px;

    position: relative;

    top: 10px;

    border-radius: 15px 15px 0px 0px;

}

#tretiy {

    margin-left: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#flex-container {

    display: flex;

    margin-left: 37px;

    align-items: center;

}

#flex-container > div {

    display: flex;

    justify-content: center;

    align-items: center;

}

#flex-container2 {

    margin-top: 10px;

    display: flex;

    margin-left: 37px;

}

#chetv {

    width: 1351.7px;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    height: 600px;

    border-top: none;

    background-color: #FFFF;

    border-radius: 15px;

}

.chetv-inner {

    width: 150px;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    text-align: center;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    padding: 10px;

    margin-top: 90px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    height: 143px;

}

#chetv .row {

    display: flex;

    justify-content: space-between;

}

.chetv-inner img {

    width: 136px;

    height: 130px;

}

.bottom-row .chetv-inner {

    margin-bottom: 120px;

    width: 166px;

    height: 50px;

    margin-top: 0px;

    background-color: #d9d9d9;

}

#combined {

    display: flex;

    flex-direction: column;

    width: 28%;

}

.card {

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: space-around;

    width: 100%;

    height: 100%;

    font-size: 23px;

    font-weight: 550;

    font-family: Inter;

    padding-left: 10px;

}

.card > div {

    width: 100%;

    display: flex;

    justify-content: space-between;

}

.card-temp {

    font-size: 30px;

}

.card-time, .card-time0 {

    color: #00bcff;

}

#weather-icon {

    max-width: 100%;

    height: auto;

}

#humid {

    padding-top: 140px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    padding-bottom: 10px;

}

hr {

    height: 1px;

    margin-top: 100px;

    margin-left: 20%;

    width: 1425px;

    background-color: #fa6901;

}

#last {

    margin-right: 20px;

    text-align: center;

    font-family: Inter;

    font-size: 16px;

    font-weight: 300;

}

}

@media (min-width: 768px) and (max-width: 1920px) {

body {

    background-color: #e0e0e0;

}

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: auto;

    margin-right: auto;

    width: 900px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

#header a {

    text-decoration: none;

    color: #fa6901;

}

#header a:hover {

    color: #e06004;

}

#header h1 {

    margin-left: 3%;

    font-size: 30px;

}

.header\_rod {

    height: 40px;

}

#header #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

}

#black\_weather {

    color: #000;

}

#navbar {

    background-color: #fa6901;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    height: 50px;

    padding: 0 10px;

    margin-left: auto;

    margin-right: auto;

    width: 883px;

    border: none;

    border-radius: 5px;

}

#navbar a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

}

#navbar input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

}

#navbar button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

}

#navbar button:hover {

    background-color: #e06004;

}

#navbar input::placeholder {

    color: #fa6901;

}

article {

    margin-left: auto;

    margin-right: auto;

    width: 900px;

}

#perviy, #vtoroy, #tretiy {

    height: 100px;

    width: 264px;

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 10px;

    box-sizing: border-box;

    background-color: #FFFF;

    overflow: hidden;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#perviy {

    margin-right: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

}

#vtoroy {

    border-bottom: none;

    height: 120px;

    position: relative;

    top: 10px; /\* Поднятие вверх \*/

    border-radius: 15px 15px 0px 0px;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#tretiy {

    margin-left: 20px;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    border-radius: 15px;

    margin-top: -30px; /\* Поднятие вверх \*/

}

#perviy img, #vtoroy img, #tretiy img {

    max-width: 100%;

    max-height: 100%;

    display: block;

}

#perviy .text, #vtoroy .text, #tretiy .text {

    overflow: hidden;

    text-overflow: ellipsis;

    white-space: nowrap;

}

#flex-container {

    display: flex;

    justify-content: center;

    align-items: center;

    margin: 20px 0;

}

#flex-container > div {

    display: flex;

    justify-content: center;

    align-items: center;

}

#flex-container2 {

    display: flex;

    justify-content: center;

    align-items: center;

    margin: 20px 0;

}

#chetv {

    width: 830px;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    height: 340px;

    border-top: none;

    background-color: #FFFF;

    border-radius: 15px;

    margin-left: auto;

    margin-right: auto;

    margin-top: -20px; /\* Поднятие вверх \*/

}

.chetv-inner {

    width: 80px;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    text-align: center;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    padding: 10px;

    margin-top: 40px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    height: 120px;

}

#chetv .row {

    display: flex;

    justify-content: space-between;

}

.chetv-inner img {

    width: 100px;

    height: 96px;

}

.bottom-row .chetv-inner {

    margin-bottom: 120px;

    width: 166px;

    height: 30px;

    margin-top: 0px;

    background-color: #d9d9d9;

}

#combined {

    display: flex;

    flex-direction: column;

    width: 28%;

}

.card {

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: space-around;

    width: 100%;

    height: 100%;

    font-size: 18px;

    font-weight: 550;

    font-family: Inter;

    padding-left: 10px;

}

.card > div {

    width: 100%;

    display: flex;

    justify-content: space-between;

}

.card-temp {

    font-size: 20px;

}

.card-time, .card-time0 {

    color: #00bcff;

}

#weather-icon {

    max-width: 100%;

    height: auto;

}

#humid {

    padding-top: 20px;

    font-size: 20px;

    font-family: Inter;

    font-weight: 500;

}

hr {

    height: 1px;

    margin-top: 20px; /\* Уменьшение отступа сверху \*/

    margin-left: auto;

    margin-right: auto;

    width: 900px;

    background-color: #fa6901;

}

#last {

    margin-right: 20px;

    text-align: center;

    font-family: Inter;

    font-size: 16px;

    font-weight: 300;

    margin-top: -20px; /\* Поднятие вверх \*/

}

}

@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&family=Inter:wght@100..900&family=Pixelify+Sans:wght@700&display=swap');

@media (min-width: 1921px) {

body {

    background-color: #e0e0e0;

}

#header {

    display: flex;

    align-items: center;

    height: 100px;

    margin-left: 20%;

    width: 1425px;

}

#header > \* {

    margin: 17px 10px 44px;

    padding: 0;

    color: #fa6901;

    font-family: Inter;

    font-size: 30px;

}

#header a {

    text-decoration: none;

    color: #fa6901;

}

#header a:hover {

    color: #e06004;

}

#header h1 {

    margin-left: 3%;

    font-size: 30px;

}

.header\_rod {

    height: 40px;

}

#header #pogoda, #karta, #novosti {

    margin-right: 50px;

    font-size: 25px;

}

#black\_weather {

    color: #000;

}

#navbar {

    background-color: #fa6901;

    border: 1px solid;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    height: 50px;

    padding: 0 10px;

    margin-left: 20%;

    width: 1405px;

    border: none;

    border-radius: 5px;

}

#navbar a {

    margin: 0 15px;

    text-decoration: none;

    color: #e0e0e0;

    font-family: Inter;

    font-size: 20px;

    font-weight: 500;

}

#navbar input {

    border-radius: 20px;

    border: 1px solid #ddd;

    height: 30px;

    text-align: center;

    color: #fa6901;

    font-size: 20px;

    outline: 0;

}

#navbar button {

    border-radius: 15px;

    height: 100%;

    background-color: #fa6901;

    border: 0;

    margin-left: 10px;

}

#navbar button:hover {

    background-color: #e06004;

}

#navbar input::placeholder {

    color: #fa6901;

}

article {

    margin-left: 20%;

}

#flex-container2 {

    margin-top: 10px;

    display: flex;

    margin-left: 37px;

}

#chetv {

    width: 1351.7px;

    display: flex;

    flex-direction: column;

    height: 800px;

    border-top: none;

    background-color: #FFFF;

    border-radius: 15px;

}

.chetv-inner {

    width: 170px;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    text-align: center;

    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

    padding: 10px;

    margin-top: 90px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    height: 170px;

    color: #fa6901;

}

#chetv .row {

    display: flex;

    justify-content: space-between;

    margin-bottom: 0;

}

.chetv-inner img {

    width: 170px;

    height: 145px;

}

.bottom-row .chetv-inner {

    margin-bottom: 0;

    width: 170px;

    height: 50px;

    margin-top: 0px;

}

#humid {

    padding-top: 160px;

    font-family: Inter;

    font-weight: 500;

    font-size: 20px;

    padding-bottom: 35px;

}

hr {

    height: 1px;

    margin-top: 100px;

    margin-left: 20%;

    width: 1425px;

    background-color: #fa6901;

}

#last {

    margin-right: 20px;

    text-align: center;

    font-family: Inter;

    font-size: 16px;

    font-weight: 300;

}

.additional-row {

    display: flex;

    justify-content: space-between;

}

.additional-row .chetv-inner {

   box-shadow: none;

    height: 30px;

    color: #000;

    margin-top: 10px;

}

.additional-row > \* {

    margin: 0;

    padding: 0;

    border: 0;

}

.cold {

    background-color: #76C43973;

}

.cool {

    background-color: #76C43999;

}

.warm {

    background-color: #76C439CC;

}

.hot {

    background-color: #75c439;

}

}

@media (min-width: 768px) and (max-width: 1920px) {

    body {

        background-color: #e0e0e0;

    }

    #header {

        display: flex;

        align-items: center;

        height: 100px;

        margin-left: auto;

        margin-right: auto;

        width: 900px;

    }

    #header > \* {

        margin: 17px 10px 44px;

        padding: 0;

        color: #fa6901;

        font-family: Inter;

        font-size: 30px;

    }

    #header a {

        text-decoration: none;

        color: #fa6901;

    }

    #header a:hover {

        color: #e06004;

    }

    #header h1 {

        margin-left: 3%;

        font-size: 30px;

    }

    .header\_rod {

        height: 40px;

    }

    #header #pogoda, #karta, #novosti {

        margin-right: 50px;

        font-size: 25px;

    }

    #black\_weather {

        color: #000;

    }

    #navbar {

        background-color: #fa6901;

        display: flex;

        justify-content: flex-start;

        align-items: center;

        height: 50px;

        padding: 0 10px;

        margin-left: auto;

        margin-right: auto;

        width: 883px;

        border: none;

        border-radius: 5px;

    }

    #navbar a {

        margin: 0 15px;

        text-decoration: none;

        color: #e0e0e0;

        font-family: Inter;

        font-size: 20px;

        font-weight: 500;

    }

    #navbar input {

        border-radius: 20px;

        border: 1px solid #ddd;

        height: 30px;

        text-align: center;

        color: #fa6901;

        font-size: 20px;

        outline: 0;

    }

    #navbar button {

        border-radius: 15px;

        height: 100%;

        background-color: #fa6901;

        border: 0;

        margin-left: 10px;

    }

    #navbar button:hover {

        background-color: #e06004;

    }

    #navbar input::placeholder {

        color: #fa6901;

    }

    article {

        margin-left: auto;

        margin-right: auto;

    }

    #flex-container2 {

        margin-top: 10px;

        display: flex;

        justify-content: center;

        align-items: center;

    }

    #chetv {

        width: 90%;

        max-width: 830px;

        display: flex;

        flex-direction: column;

        height: auto;

        border-top: none;

        background-color: #fff;

        border-radius: 15px;

        margin: 0 auto;

    }

    .chetv-inner {

        width: 90%;

        max-width: 95px;

        display: flex;

        align-items: center;

        justify-content: center;

        flex-direction: column;

        text-align: center;

        box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);

        padding: 10px;

        margin: 20px auto;

        font-family: Inter;

        font-weight: 500;

        font-size: 20px;

        height: auto;

        color: #fa6901;

    }

    #chetv .row {

        display: flex;

        justify-content: space-between;

        margin-bottom: 0;

    }

    .chetv-inner img {

        width: 100px;

        height: 96px;

    }

    .bottom-row .chetv-inner {

        margin-bottom: 0;

        width: 170px;

        height: 25px;

        margin-top: 0px;

    }

    #humid {

        padding-top: 50px;

        font-family: Inter;

        font-weight: 500;

        font-size: 20px;

        padding-bottom: 15px;

    }

    hr {

        height: 1px;

        margin-top: 100px;

        margin-left: 20%;

        width: 1425px;

        background-color: #fa6901;

    }

    #last {

        margin-right: 20px;

        text-align: center;

        font-family: Inter;

        font-size: 16px;

        font-weight: 300;

    }

    .additional-row {

        display: flex;

        justify-content: space-between;

    }

    .additional-row .chetv-inner {

        box-shadow: none;

        height: 30px;

        color: #000;

        margin-top: 10px;

        font-size: 18px;

    }

    .additional-row > \* {

        margin: 0;

        padding: 0;

        border: 0;

    }

    .cold {

        background-color: #76C43973;

    }

    .cool {

        background-color: #76C43999;

    }

    .warm {

        background-color: #76C439CC;

    }

    .hot {

        background-color: #75c439;

    }

}

ПРИЛОЖЕНИЕ Г. Листинг XML-файлов

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="footer.xsl"?>

<authorInfo>

    <surname>Мысливцев</surname>

    <name>Алексей</name>

    <faculty>ФИТ</faculty>

    <spec>ИСиТ (проф. ИС) </spec>

    <course>1-го</course>

    <group>1-ой</group>

    <topic>Веб-сайт прогноза погоды</topic>

</authorInfo>

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">

        <html>

        <head>

            <title>Информация об авторе</title>

        </head>

        <body>

            <h1>Информация об авторе</h1>

            <p>Фамилия: <xsl:value-of select="authorInfo/surname"/></p>

            <p>Имя: <xsl:value-of select="authorInfo/name"/></p>

            <p>Факультет: <xsl:value-of select="authorInfo/faculty"/></p>

            <p>Специальность: <xsl:value-of select="authorInfo/spec"/></p>

            <p>Курс: <xsl:value-of select="authorInfo/course"/></p>

            <p>Группа: <xsl:value-of select="authorInfo/group"/></p>

            <p>Тема: <xsl:value-of select="authorInfo/topic"/></p>

        </body>

        </html>

    </xsl:template>

</xsl:stylesheet>

ПРИЛОЖЕНИЕ Д. Листинг JS

import conditions from "./conditions.js";

console.log(conditions);

    const form = document.querySelector("#form");

    const input = document.querySelector("#input");

    const br = document.querySelector('#br');

    const apiKey = "868f7c09b97e48d882993602242505";

    // Функция для преобразования даты

    function formatDate(dateString) {

        const date = new Date(dateString);

        const options = { weekday: 'short', day: 'numeric', month: 'long' };

        let formattedDate = date.toLocaleDateString('ru-RU', options);

        // Преобразование первой буквы дня недели в заглавную

        formattedDate = formattedDate.charAt(0).toUpperCase() + formattedDate.slice(1);

        return formattedDate;

    }

    function formatConditionText(text) {

        return text.trimEnd();

    }

    form.onsubmit = function (event) {

        // Отменяем отправку формы

        event.preventDefault();

        const city = input.value.trim(); // обрезаем пробелы

        // Адрес запроса

        const url = `http://api.weatherapi.com/v1/forecast.json?key=${apiKey}&q=${city}&days=3&aqi=no&alerts=no`;

        // Выполнение запроса на сервер

        fetch(url).then((response) => {

            return response.json();

        }).then((data) => {

            console.log(data);

            //принимает один аргумент object, представляющий текущий элемент массива conditions, и проверяет, равен ли object.code значению переменной code.

            const info = conditions.find((object) => object.code === data.current.condition.code)

            const info2 = conditions.find((object) => object.code === data.forecast.forecastday[0].day.condition.code)

            const info3 = conditions.find((object) => object.code === data.forecast.forecastday[1].day.condition.code )

            const timeCondition = data.current.is\_day ? info.languages[23]["day\_text"] : info.languages[23]["night\_text"];

            const filePath = './pictures/' + (data.current.is\_day ? 'day' : 'night') + '/';

            const fileName = (data.current.is\_day ? info.day : info.night) + '.png';

            const imgPath = filePath + fileName;

        const imgPath2 = './pictures/day/' + formatConditionText(data.forecast.forecastday[0].day.condition.text) + '.png';

        const imgPath3 = './pictures/day/' + formatConditionText(data.forecast.forecastday[1].day.condition.text) + '.png';

        const smallImgPath = './pictures/' + (data.forecast.forecastday[0].hour[0].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[0].condition.text) + '.png';

        const smallImgPath2 = './pictures/' + (data.forecast.forecastday[0].hour[3].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[3].condition.text) + '.png';

        const smallImgPath3 = './pictures/' + (data.forecast.forecastday[0].hour[6].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[6].condition.text) + '.png';

        const smallImgPath4 = './pictures/' + (data.forecast.forecastday[0].hour[9].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[9].condition.text) + '.png';

        const smallImgPath5 = './pictures/' + (data.forecast.forecastday[0].hour[12].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[12].condition.text) + '.png';

        const smallImgPath6 = './pictures/' + (data.forecast.forecastday[0].hour[15].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[15].condition.text) + '.png';

        const smallImgPath7 = './pictures/' + (data.forecast.forecastday[0].hour[18].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[18].condition.text) + '.png';

        const smallImgPath8 = './pictures/' + (data.forecast.forecastday[0].hour[21].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[0].hour[21].condition.text) + '.png';

            const prevCard = document.querySelector("#article");

            if(prevCard) prevCard.remove();

        const html =

        `<article id="article">

        <div id="flex-container">

             <div id="perviy">

                 <div class="card">

                     <div class="card-time">Сейчас</div>

                     <div class="card-temp">${data.current.temp\_c}°C</div>

                     <div class="card-weather">${timeCondition}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath}" alt="sunny">

             </div>

             <div id="vtoroy">

                 <div class="card">

                     <div class="card-time2">${formatDate(data.forecast.forecastday[0].date)}</div>

                     <div class="card-temp">${data.forecast.forecastday[0].day.maxtemp\_c}°C</div>

                     <div class="card-weather">${info2.languages[23]["day\_text"]}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath2}" alt="sunny">

             </div>

             <div id="tretiy">

                 <div class="card">

                     <div class="card-time">${formatDate(data.forecast.forecastday[1].date)}</div>

                     <div class="card-temp">${data.forecast.forecastday[1].day.maxtemp\_c}°C</div>

                     <div class="card-weather">${info3.languages[23]["day\_text"]}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath3}" alt="sunny">

             </div>

         </div>

         <div id="flex-container2">

             <div id="chetv">

                 <div class="row top-row">

                     <div class="chetv-inner"><img src="${smallImgPath}" alt="sunny"> 0.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath2}" alt="sunny"> 3.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath3}" alt="sunny"> 6.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath4}" alt="sunny"> 9.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath5}" alt="sunny"> 12.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath6}" alt="sunny"> 15.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath7}" alt="sunny"> 18.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath8}" alt="sunny"> 21.00</div>

                 </div>

                 <div id="humid">Отноительная влажность, %</div>

                 <div class="row bottom-row">

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[0].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[3].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[6].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[9].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[12].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[15].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[18].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[21].humidity}</div>

                 </div>

             </div>

         </div>

         </article>`;

    br.insertAdjacentHTML("afterend", html);

        }).catch((error) => {

            alert('Ошибка: ' + error.message);

        });

    };

import conditions from "./conditions.js";

console.log(conditions);

    const form = document.querySelector("#form");

    const input = document.querySelector("#input");

    const br = document.querySelector('#br');

    const apiKey = "868f7c09b97e48d882993602242505";

    // Функция для преобразования даты

    function formatDate(dateString) {

        const date = new Date(dateString);

        const options = { weekday: 'short', day: 'numeric', month: 'long' };

        let formattedDate = date.toLocaleDateString('ru-RU', options);

        // Преобразование первой буквы дня недели в заглавную

        formattedDate = formattedDate.charAt(0).toUpperCase() + formattedDate.slice(1);

        return formattedDate;

    }

    function formatConditionText(text) {

        return text.trimEnd();

    }

    form.onsubmit = function (event) {

        // Отменяем отправку формы

        event.preventDefault();

        const city = input.value.trim(); // обрезаем пробелы

        // Адрес запроса

        const url = `http://api.weatherapi.com/v1/forecast.json?key=${apiKey}&q=${city}&days=3&aqi=no&alerts=no`;

        // Выполнение запроса на сервер

        fetch(url).then((response) => {

            return response.json();

        }).then((data) => {

            console.log(data);

            //принимает один аргумент object, представляющий текущий элемент массива conditions, и проверяет, равен ли object.code значению переменной code.

            const info = conditions.find((object) => object.code === data.current.condition.code)

            const info2 = conditions.find((object) => object.code === data.forecast.forecastday[0].day.condition.code)

            const info3 = conditions.find((object) => object.code === data.forecast.forecastday[1].day.condition.code )

            const timeCondition = data.current.is\_day ? info.languages[23]["day\_text"] : info.languages[23]["night\_text"];

            const filePath = './pictures/' + (data.current.is\_day ? 'day' : 'night') + '/';

            const fileName = (data.current.is\_day ? info.day : info.night) + '.png';

            const imgPath = filePath + fileName;

        const imgPath2 = './pictures/day/' + formatConditionText(data.forecast.forecastday[0].day.condition.text) + '.png';

        const imgPath3 = './pictures/day/' + formatConditionText(data.forecast.forecastday[1].day.condition.text) + '.png';

        const smallImgPath = './pictures/' + (data.forecast.forecastday[1].hour[0].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[0].condition.text) + '.png';

        const smallImgPath2 = './pictures/' + (data.forecast.forecastday[1].hour[3].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[3].condition.text) + '.png';

        const smallImgPath3 = './pictures/' + (data.forecast.forecastday[1].hour[6].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[6].condition.text) + '.png';

        const smallImgPath4 = './pictures/' + (data.forecast.forecastday[1].hour[9].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[9].condition.text) + '.png';

        const smallImgPath5 = './pictures/' + (data.forecast.forecastday[1].hour[12].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[12].condition.text) + '.png';

        const smallImgPath6 = './pictures/' + (data.forecast.forecastday[1].hour[15].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[15].condition.text) + '.png';

        const smallImgPath7 = './pictures/' + (data.forecast.forecastday[1].hour[18].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[18].condition.text) + '.png';

        const smallImgPath8 = './pictures/' + (data.forecast.forecastday[1].hour[21].is\_day ? 'day' : 'night') + '/' + formatConditionText(data.forecast.forecastday[1].hour[21].condition.text) + '.png';

            const prevCard = document.querySelector("#article");

            if(prevCard) prevCard.remove();

        const html =

        `<article id="article">

        <div id="flex-container">

             <div id="perviy">

                 <div class="card">

                     <div class="card-time0">${formatDate(data.forecast.forecastday[0].date)}</div>

                     <div class="card-temp">${data.forecast.forecastday[1].day.maxtemp\_c}°C</div>

                     <div class="card-weather">${timeCondition}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath}" alt="sunny">

             </div>

             <div id="vtoroy">

                 <div class="card">

                     <div class="card-time2">${formatDate(data.forecast.forecastday[1].date)}</div>

                     <div class="card-temp">${data.forecast.forecastday[1].day.maxtemp\_c}°C</div>

                     <div class="card-weather">${info2.languages[23]["day\_text"]}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath2}" alt="sunny">

             </div>

             <div id="tretiy">

                 <div class="card">

                     <div class="card-time">${formatDate(data.forecast.forecastday[2].date)}</div>

                     <div class="card-temp">${data.forecast.forecastday[2].day.maxtemp\_c}°C</div>

                     <div class="card-weather">${info3.languages[23]["day\_text"]}</div>

                 </div>

                 <img id="weather-icon" src="${imgPath3}" alt="sunny">

             </div>

         </div>

         <div id="flex-container2">

             <div id="chetv">

                 <div class="row top-row">

                     <div class="chetv-inner"><img src="${smallImgPath}" alt="sunny"> 0.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath2}" alt="sunny"> 3.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath3}" alt="sunny"> 6.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath4}" alt="sunny"> 9.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath5}" alt="sunny"> 12.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath6}" alt="sunny"> 15.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath7}" alt="sunny"> 18.00</div>

                     <div class="chetv-inner"><img src="${smallImgPath8}" alt="sunny"> 21.00</div>

                 </div>

                 <div id="humid">Отноительная влажность, %</div>

                 <div class="row bottom-row">

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[0].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[3].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[6].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[9].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[12].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[15].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[18].humidity}</div>

                     <div class="chetv-inner" >${data.forecast.forecastday[0].hour[21].humidity}</div>

                 </div>

             </div>

         </div>

         </article>`;

    br.insertAdjacentHTML("afterend", html);

        }).catch((error) => {

            alert('Ошибка: ' + error.message);

        });

    };

import conditions from "./conditions.js";

console.log(conditions);

const form = document.querySelector("#form");

const input = document.querySelector("#input");

const br = document.querySelector('#br');

const apiKey = "868f7c09b97e48d882993602242505";

// Функции для преобразования даты

function formatDateWithMonth(dateString) {

    const date = new Date(dateString);

    const options = { weekday: 'short', day: 'numeric', month: 'long' };

    let formattedDate = date.toLocaleDateString('ru-RU', options);

    // Преобразование первой буквы дня недели в заглавную

    formattedDate = formattedDate.charAt(0).toUpperCase() + formattedDate.slice(1);

    // Разделение строки на две части: день недели и дата

    const [weekday, day, month] = formattedDate.split(' ');

    return `${weekday}\n${day} ${month}`;

}

function formatDateWithoutMonth(dateString) {

    const date = new Date(dateString);

    const options = { weekday: 'short', day: 'numeric' };

    let formattedDate = date.toLocaleDateString('ru-RU', options);

    // Преобразование первой буквы дня недели в заглавную

    formattedDate = formattedDate.charAt(0).toUpperCase() + formattedDate.slice(1);

    // Разделение строки на две части: день недели и день

    const [weekday, day] = formattedDate.split(' ');

    return `${weekday}\n${day}`;

}

function formatConditionText(text) {

    return text.trimEnd();

}

function getTemperatureClass(temp) {

    if (temp <= 0) {

        return 'cold';

    } else if (temp <= 15) {

        return 'cool';

    } else if (temp <= 25) {

        return 'warm';

    } else {

        return 'hot';

    }

}

form.onsubmit = function (event) {

    // Отменяем отправку формы

    event.preventDefault();

    const city = input.value.trim(); // обрезаем пробелы

    // Адрес запроса

    const url = `http://api.weatherapi.com/v1/forecast.json?key=${apiKey}&q=${city}&days=7&aqi=no&alerts=no`;

    // Выполнение запроса на сервер

    fetch(url).then((response) => {

        return response.json();

    }).then((data) => {

        console.log(data);

        const maxTemps = data.forecast.forecastday.map(day => day.day.maxtemp\_c);

        const minTemps = data.forecast.forecastday.map(day => day.day.mintemp\_c);

        const getImagePath = (dayIndex) => `./pictures/day/${formatConditionText(data.forecast.forecastday[dayIndex].day.condition.text)}.png`;

        const dates = data.forecast.forecastday.map(day => day.date);

        const prevCard = document.querySelector("#article");

        if (prevCard) prevCard.remove();

        let html = `

        <article id="article">

            <div id="flex-container2">

                <div id="chetv">

                    <div class="row top-row">

                        <div class="chetv-inner">${formatDateWithMonth(dates[0])} <img src="${getImagePath(0)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[1])}<img src="${getImagePath(1)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[2])}<img src="${getImagePath(2)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[3])} <img src="${getImagePath(3)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[4])} <img src="${getImagePath(4)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[5])}<img src="${getImagePath(5)}" alt="sunny"></div>

                        <div class="chetv-inner">${formatDateWithoutMonth(dates[6])} <img src="${getImagePath(6)}" alt="sunny"></div>

                    </div>

                    <div id="humid">Температура воздуха, °C</div>

                    <div class="row additional-row">

                        ${maxTemps.map(temp => `<div class="chetv-inner">${temp}°C</div>`).join('')}

                    </div>

                    <div class="row bottom-row">

                        ${maxTemps.map(temp => `<div class="chetv-inner ${getTemperatureClass(temp)}" style="color: transparent;">${temp}°C</div>`).join('')}

                    </div>

                    <div class="row additional-row">

                        ${minTemps.map(temp => `<div class="chetv-inner">${temp}°C</div>`).join('')}

                    </div>

                </div>

            </div>

        </article>`;

        br.insertAdjacentHTML("afterend", html);

    }).catch((error) => {

        alert('Ошибка: ' + error.message);

    });

};