**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc184983821)

[Глава 1. Теоретическая часть 3](#_Toc184983822)

[1.1. Html 4](#_Toc184983823)

[1.2. Flexbox 5](#_Toc184983824)

[1.3. Class and Id 6](#_Toc184983825)

[1.4. Свойства в CSS 8](#_Toc184983826)

[Глава 2. Практическая часть 10](#_Toc184983827)

[2.1. Создание проекта 11](#_Toc184983828)

[2.2. Загрузка библиотеки: 13](#_Toc184983829)

[ЗАКЛЮЧЕНИЕ 16](#_Toc184983830)

# ВВЕДЕНИЕ:

В современном мире интернет-магазины стали неотъемлемой частью нашей повседневной жизни, предоставляя пользователям удобный доступ к разнообразным товарам и услугам. Одной из популярных ниш в электронной коммерции является продажа книг, особенно манги, манхвы и маньхуа, которые привлекают внимание как молодежной аудитории, так и более взрослым читателям.   
  
Создание эффективного и привлекательного интернет-магазина требует не только качественного контента, но и продуманного дизайна, который обеспечивает удобство навигации и положительный пользовательский опыт. В этой работе мы сосредоточимся на использовании технологий HTML и CSS с применением Flexbox для разработки интерфейса интернет-магазина, специализирующегося на продаже книжной манги, манхвы и маньхуа.

Flexbox — это мощный инструмент для создания адаптивных и гибких макетов, который позволяет легко управлять расположением элементов на странице. Использование Flexbox в нашем проекте позволит создать современный и отзывчивый дизайн, обеспечивающий оптимальное отображение на различных устройствах. Мы рассмотрим основные принципы работы с Flexbox.

В современном мире интернет-магазины стали неотъемлемой частью нашей повседневной жизни, предоставляя пользователям удобный доступ к разнообразным товарам и услугам. Одной из популярных ниш в электронной коммерции является продажа книг, особенно манги, манхвы и маньхуа, которые привлекают внимание как молодежной аудитории, так и более взрослым читателям.

# Глава 1. Теоретическая часть

В начале расскажу немного про сам веб-сайт . Потом мы разберем из чего состоит данный сайт, что такое Flexbox ,и многое другое что связана с html и css.

Веб-сайт на HTML (HyperText Markup Language) — это набор связанных веб-страниц, которые создаются с использованием языка разметки HTML. HTML является основным языком для создания и структурирования контента в Интернете.Основной целью моего веб-сайта является предоставление пользователям удобного интерфейса для покупки книг.

Удобства в моем сайте:

Удобство покупок: Обеспечение простоты процесса покупки с различными способами оплаты и доставки.

Информационный контент: Публикация статей и новостей о новинках, событиях в мире манги и манхвы.

Аудитория:  
Целевая аудитория включает подростков и молодых людей в возрасте от 12 до 30 лет, интересующихся японской и восточной культурой, а также любителей комиксов и графических новелл. Также стоит учитывать взрослых читателей, которые могут быть заинтересованы в коллекционных изданиях.  
  
Структура сайта:

1. Главная страница: Привлекательный дизайн с акцентом на новинки и популярные товары. Блоки для акций и специальных предложений.  
2. Каталог: Разделение на категории (манга, манхва, маньхуа) с возможностью фильтрации по жанрам (приключения, романтика, фэнтези и др.).  
3. Страница товара: Подробное описание книги, обложка, рецензии пользователей, возможность добавления в корзину.

## Html:

**HTML** (от английского HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

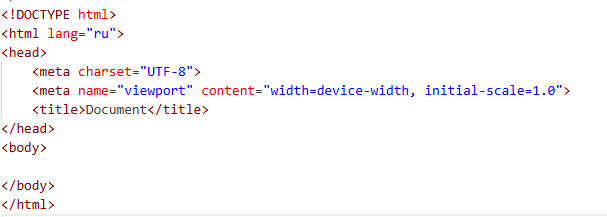


Рисунок 1.1.1 – Простейший html

Для создания документа HTML5 нам нужны в первую очередь два элемента:

DOCTYPE и html. Элемент doctype или Document Type Declaration сообщает веб-браузеру тип документа. <!DOCTYPE html> указывает, что данный документ является документом html и что используется html5, а не html4 или какая-то другая версия языка разметки.

А элемент html между своим открывающим и закрывающим тегами содержит все содержимое документа.

Внутри элемента html мы можем разместить два других элемента: head и body. Элемент head содержит метаданные веб-страницы - заголовок веб-страницы, тип кодировки и т.д., а также ссылки на внешние ресурсы - стили, скрипты, если они использутся. Элемент body собственно определяет содержимое html-страницы.

В элементе head определено два элемента:

элемент title представляет заголовок страницы

элемент meta определяет метаинформацию страницы. Для корректного отображения символов предпочтительно указывать кодировку. В данном случае с помощью атрибута charset="utf-8" указываем кодировку utf-8.

## Flexbox:

Flexbox - это общее название для модуля Flexible Box Layout, который имеется в CSS3. Данный модуль определяет особый режим компоновки/верстки пользовательского интерфейса, который называется flex layout. В этом плане Flexbox предоставляет иной подход к созданию пользовательского интерфейса, который отличается от табличной или блочной верстки.

В данном модуле хранится следующий список свойств:

Свойства Flexbox:

Контейнерные свойства:

* display: flex или display: inline-flex: Переводит элемент в режим Flexbox.
* flex-direction: Определяет основную ось (горизонтальную или вертикальную) для размещения дочерних элементов.
* flex-wrap: Разрешает перенос элементов на следующую строку (или столбец) в зависимости от направления оси.
* justify-content: Выравнивает элементы по главной оси.
* align-items: Выравнивает элементы по поперечной оси.
* align-content: Выравнивает линии элементов по поперечной оси (если flex-wrap установлен в wrap).
* Свойства дочерних элементов:
* order: Определяет порядок дочерних элементов вдоль главной оси.
* flex-grow: Устанавливает, насколько элемент будет расширяться по главной оси при наличии свободного пространства.
* flex-shrink: Устанавливает, насколько элемент будет сжиматься по главной оси при нехватке пространства.
* flex-basis: Устанавливает начальную ширину или высоту элемента.
* Другие свойства:
* flex: Сокращенное свойство, которое устанавливает одновременно flex-grow, flex-shrink и flex-basis.
* gap: Устанавливает расстояние между элементами как по главной, так и по поперечной оси.
* GetHttpContext() – возвращает объект HttpContext для текущего подключения в случае если для подключение используется HTTP протокол. В ином случае null.

Вот пример Flexbox(Рисунок 1.2.1):

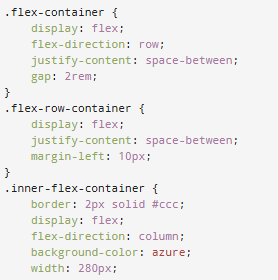


Рисунок 1.2.1 – Пример использования Flexbox

## Class and Id:

Классы:

Классы используются для применения стиля к группе элементов с общими характеристиками.

К одному элементу можно применить несколько классов.

Классы определяются в CSS с использованием точки ., например: .my-class.

Классы позволяют легко применять один и тот же стиль к нескольким элементам, не дублируя код CSS.

Идентификаторы (ID):

Идентификаторы используются для определения уникального элемента на странице.

К одному элементу можно применить только один идентификатор.

Идентификаторы определяются в CSS с использованием решетки #, например: #my-id.

Идентификаторы полезны для применения стиля к конкретному элементу, например, к форме или навигационному меню.

Используйте классы, когда вам нужно применить стиль к группе элементов с общими характеристиками. Например, вы можете создать класс для всех заголовков на странице или для всех ссылок в определенном разделе.

Используйте идентификаторы, когда вам нужно применить стиль к конкретному элементу. Например, вы можете использовать идентификатор для стилизации формы входа или навигационного меню.

Использование классов и идентификаторов позволяет организовать и управлять стилями на странице, что делает код более гибким и простым в обслуживании. Пример работы с Классами (Рисунок 1.3.1), Пример работы с Идентификаторами (Рисунок 1.3.2)

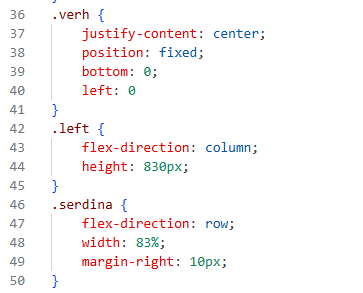


Рисунок 1.3.1 – Пример работы с Классами

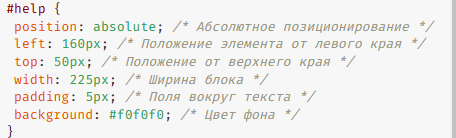


Рисунок 1.3.2 – Пример работы с Идентификаторами

## Свойства в CSS:

CSS позволяет тегам HTML назначать любые визуальные стили, формируя таким образом уникальный внешний вид сайта. Делается это при помощи свойств CSS.

Границы и фон:

* border: Добавляет рамку вокруг элемента.
* border-width: Устанавливает ширину рамки.
* border-color: Устанавливает цвет рамки.
* border-style: Устанавливает стиль рамки (сплошная, пунктирная, двойная и т.д.).
* background-color: Задает цвет фона элемента.
* background-image: Устанавливает фоновое изображение.
* background-position: Устанавливает положение фонового изображения.
* background-repeat: Устанавливает, как фоновое изображение будет повторяться.
* background-size: Устанавливает размер фонового изображения.
* Размеры и позиционирование:
* width/height: Задает ширину и высоту элемента.
* max-width/max-height: Устанавливает максимальную ширину и

высоту элемента.

* min-width/min-height: Устанавливает минимальную ширину и высоту элемента.
* margin: Устанавливает внешние отступы элемента.
* padding: Устанавливает внутренние отступы элемента.
* position: Устанавливает положение элемента (статическое, абсолютное, относительное, фиксированное).
* top/bottom/left/right: Устанавливают положение элемента относительно его родительского элемента.
* z-index: Устанавливает порядок слоев для перекрывающихся элементов.

Шрифты и текст:

* color: Задает цвет текста.
* font-family: Устанавливает шрифт текста.
* font-size: Устанавливает размер шрифта.
* font-weight: Устанавливает толщину шрифта.
* text-align: Выравнивает текст по горизонтали.
* text-decoration: Устанавливает подчеркивание, зачеркивание или перечеркивание текста.
* text-transform: Преобразует текст в верхний, нижний или заглавные буквы.

Другие свойства:

* display: Устанавливает тип отображения элемента (блок, строчный,скрытый и т.д.).
* visibility: Устанавливает видимость элемента.
* opacity: Устанавливает прозрачность элемента.
* cursor: Устанавливает курсор мыши при наведении на элемент.



Рисунок 1.5.1 –Пример использования css свойств.

# 

# Глава 2. Практическая часть

## Создание проекта:

Для начала необходимо создадать проект. Для создания и написания backend мы будем использовать одну из популярных IDE Visual Studio Code 2024.

В начале открываем IDE(Рисунок 2.1.1).

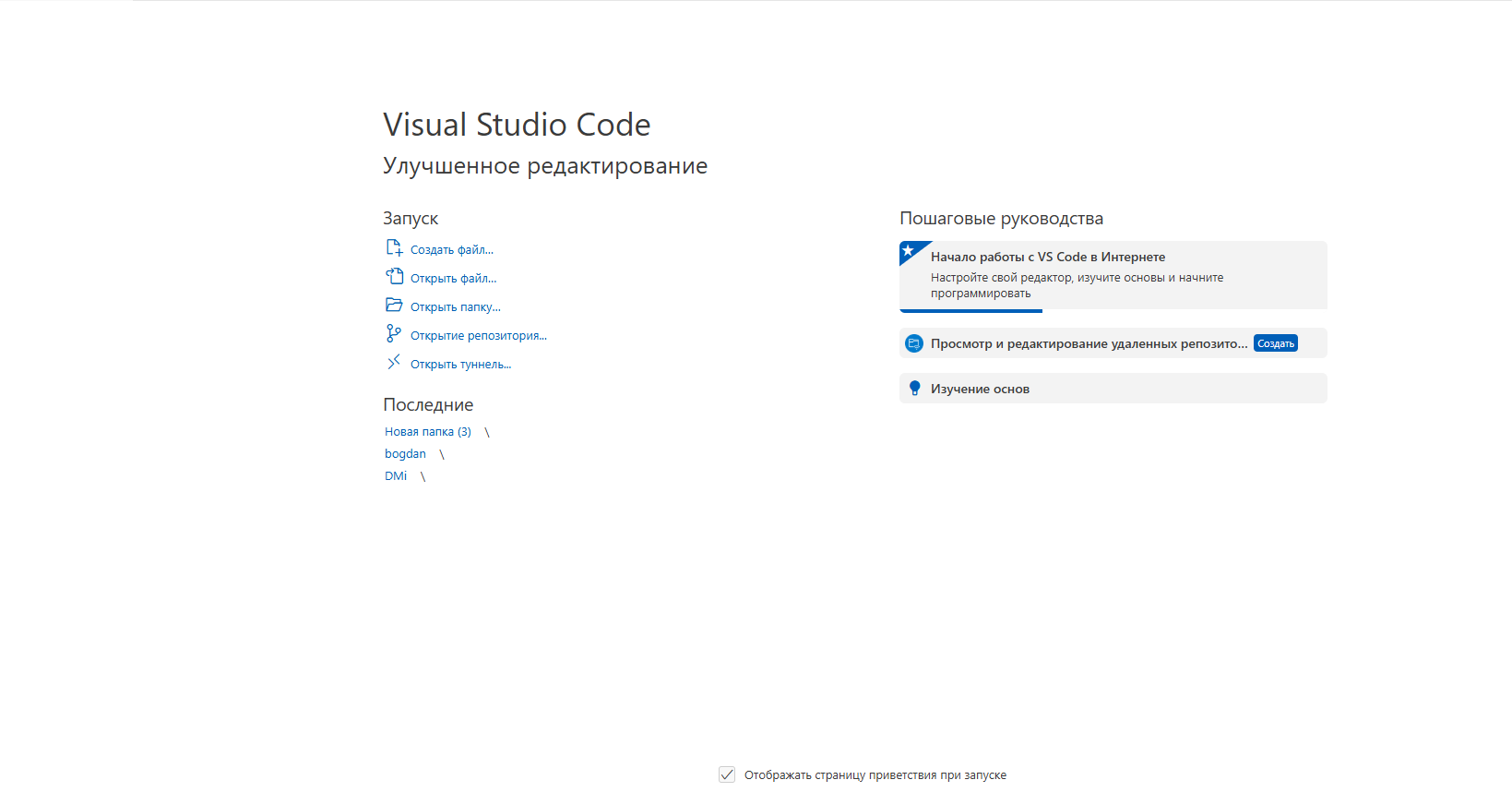


Рисунок 2.1.1 – Visual Studio Code 2024

Далее создадим проект. Выбираем тип проекта, указываем название и тип файла(Рисунок 2.1.2).

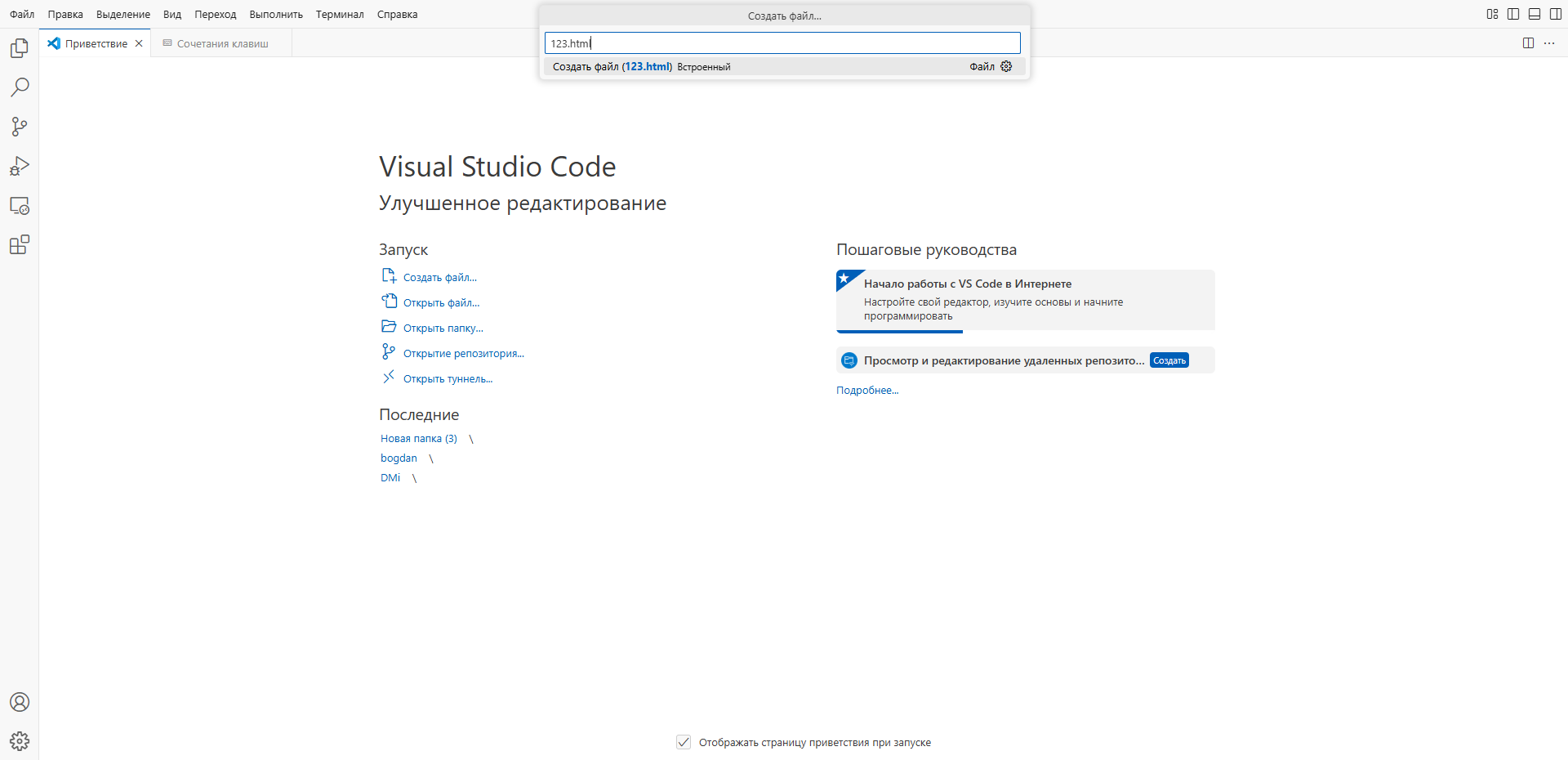


Рисунок 2.1.2 – настройка нового проекта

Затем необходимо указать путь где будет располагаться файл(Рисунок2.1.3).

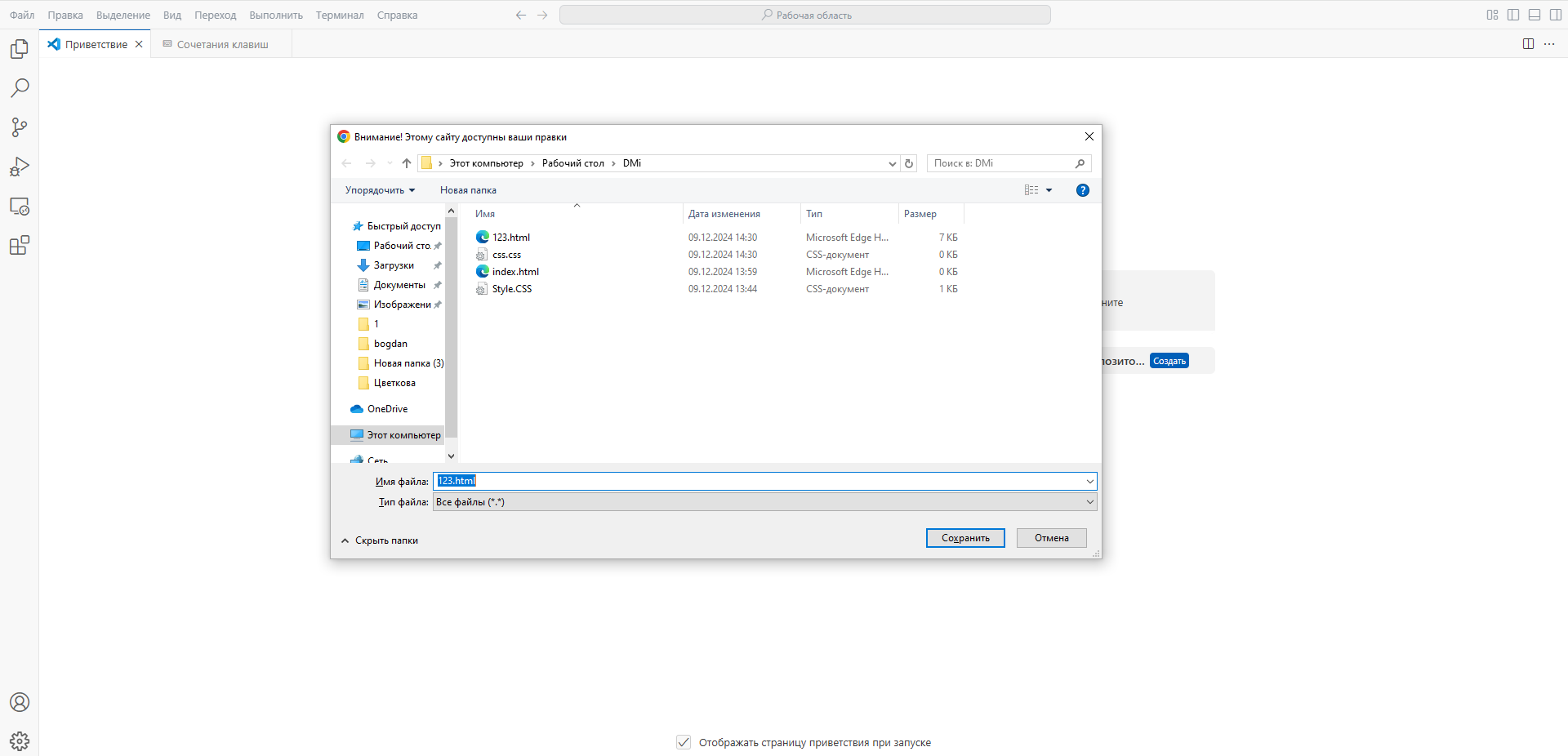


Рисунок 2.1.3

По итогу у нас должно открыться следующее окно, указывающее на успешное создание проекта(Рисунок 2.1.4).

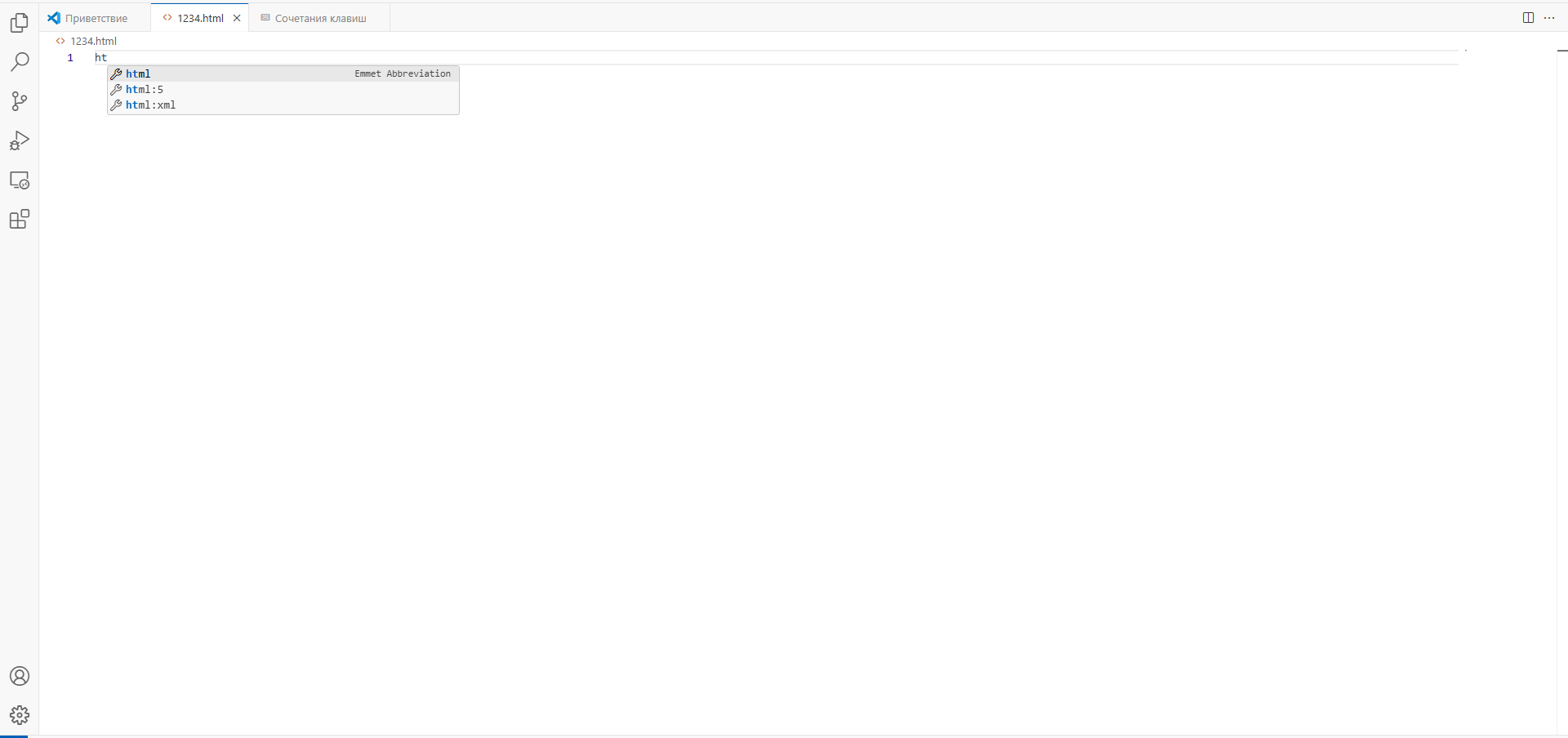


Рисунок 2.1.4 – окно успешно созданного проекта

## Загрузка библиотеки:

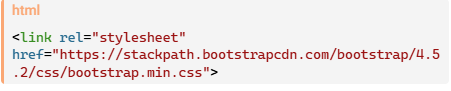
Для загрузки библиотек в HTML и CSS вы можете использовать несколько подходов, в зависимости от того, какие именно библиотеки вам нужны. Вот основные способы подключения популярных библиотек:

**Подключение CSS-библиотек:**

**a. Bootstrap**

Bootstrap — это популярный фреймворк для разработки адаптивных

веб-сайтов.



Подключение CSS-библиотеки **Bootstrap**

**b. Font Awesome**

Font Awesome : предоставляет иконки и шрифты.



**Подключение JavaScript-библиотек:**

**a. jQuery**

jQuery — это библиотека JavaScript, которая упрощает работу с DOM.



**Подключение JavaScript-библиотек** jQuery

**b. Bootstrap JS**

Если вы используете Bootstrap, вы также можете подключить его JavaScript.



**Подключение JavaScript-библиотек** Bootstrap

**3. Пример HTML-документа**  
Вот пример простого HTML-докумена с подключением библиотек: 

**Пример** с подключением библиотек в простой HTML-документ

**4. Дополнительные библиотеки**

В зависимости от ваших потребностей, вы можете подключать и другие библиотеки:  
**Animate.css** для анимаций:



**Normalize.css** для сброса стилей:



# ЗАКЛЮЧЕНИЕ:

В заключение, разработка приложений для веб-конференций представляет собой важный шаг в эволюции коммуникационных технологий, особенно в условиях современного мира, где удаленная работа и глобализация становятся нормой. Использование таких технологий, как Flexbox, позволяет обеспечить надежную и эффективную работу сайта, что значительно улучшает качество взаимодействия и обмена информацией. В данной работе мы подробно рассмотрели процесс создания сайта, что не только углубляет понимание технических аспектов разработки, но и предоставляет читателям необходимые знания для создания собственных решений в этой области. Таким образом, данная работа служит не только теоретическим, но и практическим руководством для всех, кто стремится освоить создание современных сайтах.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

<https://webref.ru/css/selector/id>

[https://metanit.com/](https://metanit.com/web/html5/12.1.php)

<https://www.programiz.com/html/online-compiler/>

<https://skillbox.ru/media/code/chto_takoe_html/>

<https://www.w3schools.com/>)

<https://developer.mozilla.org/en-US/docs/Web>)  
<https://css-tricks.com/>)  
<https://www.freecodecamp.org/><https://www.codecademy.com/>[HTML.com](https://html.com/)](<https://html.com/><https://www.w3schools.com/><https://developer.mozilla.org/en-US/docs/Web><https://css-tricks.com/><https://www.freecodecamp.org/><https://www.codecademy.com/><https://html.com/><http://www.csszengarden.com/><https://www.smashingmagazine.com/><https://alistapart.com/><https://www.tutorialspoint.com/html/index.htm><https://www.geeksforgeeks.org/html/><https://stackoverflow.com/><https://cssreference.io/><https://learn-css.org/><https://htmldog.com/><https://web.dev/><https://caniuse.com/><https://css-tricks.com/almanac/><https://designmodo.com/><https://www.sitepoint.com/><https://www.frontendmentor.io/><https://codepen.io/><https://csslayout.io/><http://html5doctor.com/><https://www.dummies.com/category/subjects/internet-41335/><https://tutsplus.com/><https://www.coursera.org/><https://www.udacity.com/><https://www.edx.org/><https://pages.github.com/>