Министерство образования и науки Российской Федерации

ФГАОУ ВО «Севастопольский государственный университет»

Институт радиоэлектроники и информационной безопасности

Кафедра радиоэлектроники и телекоммуникаций

**ОТЧЁТ**

по лабораторной работе №2

«**ИСПОЛЬЗОВАНИЕ CSS ПРИ**

**РАЗРАБОТКЕ WEB -САЙТА**»

по дисциплине

«Коммуникации в сети интернет и принципы построения сайтов»

Выполнил: студент гр. РС/с-19-1-о

Дытынчук Александр Михайлович

Вариант №11

Защитил с оценкой: \_\_\_\_\_\_

Принял: ассистент Табакаев Д. И.

Севастополь

2020

**СОДЕРЖАНИЕ**

1. Описание лабораторной работы.............................................................................3

1.1. Цель работы..........................................................................................................3

1.2. Индивидуальное задание.....................................................................................3

1.3. Теоретические сведения......................................................................................3

2. Выполнение лабораторной работы......................................................................16

2.1 Структура разработанного сайта....................................................................... 16

2.2. Текст программы................................................................................................16

2.3. Таблица свойств CSS стилей использованные для создания верстки документа...................................................................................................................27

2.4. Скриншоты выполнения программы…………………..……………………..28

Выводы.......................................................................................................................31

**1. ОПИСАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ**

**1.1. Цель работы**

Научиться работать с каскадными таблицами стилей, а также изучить

основные методы работы с селекторами и построения сайта блочной

верстки.

**1.2. Индивидуальное задание**

Создать сайт, состоящий и нескольких веб - страниц согласно

варианту, который выдается преподавателем.

**1.3. Теоретические сведения**

1.3.1. Назначение стилевых таблиц

Собственные средства HTML (теги и их атрибуты) выполняют две

основные роли: поддержку структуры документа (состав и взаимосвязи

элементов) и определение внешнего вида визуальных элементов. Идея

разделения описания внешнего вида документа от элементов,

определяющих его структуру, воплотилась в технологии, называемой

каскадными таблицами стилей (Cascading Style Sheets, CSS). Таблица

стилей, действует подобно шаблону форматирования,может быть

разработана отдельно от HTML-документа, а затем применена к нему.

Изменяя содержимое таблицы стилей, можно изменять внешний вид

HTML-документов, не затрагивая их структуры информационного

содержания. Одна и та же таблица стилей может применяться к

нескольким документам, и, наоборот, к одному и тому же документу

может быть применено несколько таблиц стилей. В последнем случае

браузер учитывает приоритеты таблиц и по определенным правилам

разрешает возникающие конфликты, в результате чего таблицы

выстраиваются неким каскадом.

Кроме технологичности стилизации HTML-документов, CSS

обеспечивают еще две важные вещи: произвольное позиционирование

элементов и создание визуальных эффектов, таких как полупрозрачность и

трансформации графических изображений и текстов.

1.3.2. Встраивание таблиц стилей в HTML-документ

Для применения каскадной таблицы стилей к HTML-документу

необходимо ее связать с ним или встроить в него. Это можно сделать

четырьмя способами:

1. Вставка непосредственно в заголовок HTML-документа. Правила

таблицы стилей заключаются в контейнерный тег <style>;

2. Вставка непосредственно тег виде строки описания в атрибуте Style;

3. Импорт - вставка таблицы стилей из внешнего файла. Файл таблицы

стилей является текстовым файлом с расширением css. Оператор @import

используется перед другими правилами таблицы стилей в контейнере

<style> или в css-файле.

4. Связывание с таблицей стилей в внешнем файле с помощью ссылки

задаваемой тегом <link>, который помещаемого в контейнер <head>.

Таблица стилей, вставленная с помощью тега <style>, действует на

элементы только текущего HTML-документа, в котором этот тег

находится. Если ту же таблицу необходимо применить и к другим

документам, то ее код придется повторить в соответствующих HTML-страницах. При этом возрастает общий объем файлов сайта, а также

трудозатраты в случае необходимости изменить его стиль. Чтобы избежать

этого, используют импорт или связывание таблиц из внешних css-файлов.

Когда требуется изменить параметры стилей для отдельных элементов

(например, их оформление), используют атрибут style. Возможно также

комбинирование всех способов встраивания таблиц.

В записях таблиц стилей можно задавать комментарии, которые задаются

символами /\* и \*/.

В тег <style> имеет следующие атрибуты:

type - для каскадных таблиц стилей всегда имеет значение text/css;

В media - определяет тип устройства вывода. Браузеры обычно используют

следующие значения: screen (экран), print (печать) и all (все). Можно

создать стили отдельно для отображения документа на экране монитора и

для вывода на печать.

1.3.3. Типы селекторов

CSS состоит из правил, а каждое правило – из селектора и блока

объявлений. Блок объявлений содержит CSS-свойства, определяющие

отображение элемента веб-страницы в браузере. Селектор отвечает за

выбор этого самого элемента. Селектор служит для однозначной

идентификации HTML элемента средствами CSS. Он позволяет выбирать

именно тот элемент (или группу элементов), который нужен. С помощью

простых селекторов можно выбирать:

― все объекты – универсальный селектор;

― объекты определенного типа;

― объекты с заданным классом;

― объект с определенным идентификатором;

― объекты с определенными характеристиками – селекторы

атрибутов.

Объединяя простые селекторы можно выбирать объекты по более

сложным правилам:

― объекты, находящиеся внутри какого-то объекта – селектор

потомка;

― объекты, непосредственно вложенные в какой-то объект –

дочерний селектор;

― объект, расположенный после другого объекта – сестринский селектор.

Также существуют селекторы псевдоклассов и псевдоэлементов. Они

позволяют назначать стили элементам, которые зависят не только от

разметки, но и от состояния документа.

1.3.4. Универсальный селектор

Предназначен для выбора всех элементов. Стили, указанные для

универсального селектора применяются ко всем элементам сразу.

Обычно применяется для сброса зависящих от браузера начальных

значений стилей (в частности, отступов).

1.3.5 Селектор типа

Предназначен для выбора всех элементов определенного типа. Стили

применяются ко всем элементам указанного типа не зависимо от уровня

вложенности. Применяется для задания общих, для всех элементов

определенного типа, стилей.

1.3.6. Селектор класса

Предназначен для выбора всех элементов по имени класса (по

значению атрибута class). Стили применяются к любым тегам с

соответствующим классом. Важно учитывать, что в отличие от названий

HTML-тегов, в названиях классов различаются большие и маленькие

буквы. То есть class="active" и class="Active" – это совсем разные классы.

HTML позволяет задавать в качестве значения атрибута class список

разделенных пробелами названий (порядок следования не имеет значения).

Другими словами, один элемент может иметь сразу несколько классов.

Стили, относящиеся к каждому из перечисленных классов, будут,

объединяясь, применяться к этому элементу. IE6 не понимает такой

записи.

Пример. Правило распространяется только на элементы, в списке

классов которых встречаются (в любом порядке) и класс popup и класс

active.

Концепция классов, наиболее часто применяется при верстке.

Например, верстать страницу, используя в качестве контейнеров теги div,

задавая им определенные классы (в соответствии с функциональным

назначением).

1.3.7. Селектор идентификатора

Предназначен для выбора элемента по уникальному идентификатору

(значению атрибута id). Позволяет задать стили конкретному HTML-элементу.

Комбинируя селектор идентификатора и селектор типа, можно

объединить их свойства. Учитывая, что id на странице должен быть

уникальным, смысла в таком комбинировании не много. Как правило,

идентификаторы применяются там, где предполагается работа скриптов.

Например, в формах или в каких-то динамических элементах. Так же с

помощью id можно подчеркнуть уникальность элемента, его присутствие в

единственном экземпляре.

1.3.8. Единицы измерения

Единицы измерения CSS используются для указания размеров

различных элементов. Есть абсолютные и относительные единицы

измерения. Абсолютные единицы не зависят от устройства вывода, а

относительные единицы определяют размер элемента относительно

значения размера, используемого в родительском элементе. Ниже

представлена таблица и подробное описание каждой единицы измерения

CSS.

Абсолютные единицы:

Миллиметр - mm, сантиметр - cm и дюйм – in. Само собой

разумеется, что это абсолютные единицы измерения. Один cm = 0.39370in,

1in = 2.54cm и 10mm = 1cm. Компьютерные дисплеи плохо вычисляют

данные единицы измерения, таким образом, у этих величин ограниченное

применение и обычно их используют при указании размера для вывода

страниц на печать.

Пиксели – px. Пиксель это маленькая точка на экране. Пиксели

определяют размер элемента. Использование пикселей дает вам точный

контроль над размером элемента, позволяя вам точно вычислить его

ширину и высоту, это будет полезным для точной разметки дизайна

страницы. Однако, есть несколько минусов использования пикселей, вы

должны быть осведомлены о них:

Во-первых, установка размера шрифта с помощью пиксельных единиц

не позволяет пользователю изменять размер шрифта с помощью настроек в

браузере. Если размер шрифта 12 пикселей, он всегда будет иметь высоту

12 точек, независимо от того, что пользователь установил размер шрифта

по умолчанию в браузере. Так что, если вы решили указать размер

шрифтов, следует подумать об использовании другой единицы измерения.

Во-вторых, когда речь заходит о печатных средствах массовой

информации, пиксели не имеют реального значения. При разработке

документа для печати, устройство печати должно будет само догадываться

о том, что вы имели ввиду с точки зрения физических размеров. Хотя

обычно можно просмотреть документ и внести изменения перед печатью

Точки - pt и пики – pc. Эти единицы измерения чаще всего используются

для указания именно размера шрифта, например, в обычном блокноте, мы

часто указываем размер текста равный 14, это значение как раз и есть

единицы измерения pt.

Точка (1pt) равна 1/72 дюйма, в то время как пика (1pc) равна 1/6

дюйма (1pc = 12pt). Документы, предназначенные для печати, будут иметь

возможность сообщить устройству именно тот размер шрифта, который

следует использовать при печати. Цифровым дисплеям, однако, придется

самим догадываться, как конвертировать эти единицы

в пиксели, и нет никаких реальных универсальных способов узнать, как

это будет сделано. Поскольку точки (pt) были использованы с первых дней

Интернета, большинство браузеров автоматически могут установить

соотношения между пикселями и точками, но это по сути неправильно.

Помните, что небольшие дисплеи сегодня могут иметь высокое

разрешение, так что определить то, насколько большой «дюйм на экране»

будет практически невозможно на устройствах. Для указания размера

элементов на веб-страницах следует избегать использования этих единиц.

Относительные единицы:

― Процент - %. Самая простая единица измерения, это процент (%),

она не имеет напрямую никакого отношения к размеру шрифта

или элемента в целом и может быть использована в комбинации с

другими единицами измерения, указывающими величину. Размер

установленный в процентах, напрямую зависит от размера

родительского элемента, например, размер шрифта задается

относительно размера шрифта родительского элемента, также

высота и ширина выражается в процентах относительно высоты и

ширины родительского элемента;

― Вычисляемая x-высота – ex. Единица измерения ex используется

достаточно редко. В качестве основы для размера 1ex используется

высота символа "x" в нижнем регистре указанного шрифта.

Большинство браузеров не поддерживает эту единицу измерения

должным образом, и она не рекомендуется для использования в

документах, предназначенных для браузеров;

― Вычисляемая единица – em. Em является относительной

единицей измерения. Один em равен 16px. Если em используется

для определенного элемента, то за 1em принимается размер

шрифта его родителя.

1.3.9. Описание шрифтов

Для описания шрифтов с помощью CSS применяются следующие

стилевые параметры:

1. font-family - определяет список семейств шрифтов. Обычно задаются

несколько похожих шрифтов в порядке предпочтения на случай, если в

компьютере пользователя нет нужного шрифта при указании имени

группы шрифтов, как показано выше, браузер подбирает подходящий для

отображения шрифт данной группы из имеющегося набора шрифтов. Если

название шрифта состоит из нескольких слов, то оно заключается в

кавычки. Список шрифтов желательно завершить родовым именем

шрифта: serif, sans-serif, cursive, fantasy или monospace. Например, для

шрифта Times родовым является serif, для Helvetica – sans-serif, для Courier

– monospace.

2. font-size – размер шрифта. Значение может быть задано различными

способами:

− абсолютный размер, задаваемый с помощью ключевых слов: xx-small,

x-small, medium, large, x-large, xx-large. Эти значения представляют

индексы таблицы размеров шрифтов, поддерживаемой браузером. По

умолчанию используется значение medium (средний);

− относительный размер, задаваемый с помощью ключевых слов: larger

(больше) и smaller (меньше). Эти значения интерпретируются

относительно таблицы размеров шрифтов браузера и размера шрифта

элемента-родителя. Например, если элемент-родитель имеет шрифт

размером medium, то значение larger для элемента-потомка сделает размер

шрифта равным large (большой). В CSS1 масштабирующий множитель

равен 1,5, в CSS2 – 1,2. Размер шрифта относительно элемента-родителя

можно задавать и в процентах;

− размер, задаваемый в абсолютных единицах длины (обычно

pt=0,35мм).

3. font-weight – жирность шрифта. Возможные значения: normal, bold

(жирный), bolder (жирнее), lighter (светлее) или одним из девяти целых

чисел от 100 до 900. Ключевому слову normal соответствует числовое

значение 400; ключевое слово bold задает обычный жирный (полужирный)

шрифт и соответствует числовому значению 700. Следует иметь в виду,

что в текущем семействе может и не быть шрифта с заданной степенью

жирности. В этом случае можно лишь гарантировать, что шрифт с

большим значением параметра font-weight будет не светлее, чем шрифт с

меньшим значением этого параметра.

4. font-style – стиль шрифта. Возможные значения: normal (нормальный

прямой), italic (курсив) и oblique (наклонный). По умолчанию применяется

значение normal. Если уже имеется готовый шрифт, соответствующий

заданному стилю, то он будет применен. В противном случае текущий

шрифт будет изменен программным способом.

5. font-variant – вариант стиля шрифта. Возможные значения: normal

(принимается по умолчанию) и small-caps. Значение normal не влияет на

отображение шрифта. Значение small-caps заменяет строчные буквы

прописными, но делает их несколько меньшими по размеру, чем

прописные буквы текущего шрифта.

6. font – позволяет установить сразу несколько свойств шрифта в

одном определении: font-style, font-variant, font-weight, font-size, line-height,

font-family. Значения этих параметров указываются через пробел в том

порядке, в котором они были перечислены. Допустимо не указывать

первые три параметра, что соответствует значению normal, принятому для

них по умолчанию. Если задается высота строки (line-height), то ее

значение отделяется от размера шрифта (font-size) прямым слэшем (/). Если

список семейств шрифтов (font-family) содержит более одного элемента, то

последние отделяются друг от друга запятыми.

1.3.10 Задание цвета и фона

Цвет текста определяется с помощью параметра color, а цвет фона

элемента – с помощью параметра background-color. По умолчанию цвет

фона элемента определен как прозрачный (transparent). Значения этих

параметров задаются именем цвета или числовым представлением в

системе RGB.

Числовое представление цвета допускает следующие варианты:

― шестнадцатеричное число (#ffff00);

― тройка целых десятичных чисел, каждое из которых представляет

яркость одной из RGB-составляющих цвета в диапазоне от 0 до

255. (rgb(255,255,0));

― тройка вещественных чисел со знаками %, каждое из которых

представляет яркость одной из RGB-составляющих цвета в

диапазоне от 0 до 100%. rgb(100%, 100%, 0%).

В качестве фона элемента или всего документа, можно использовать

изображение из файла по указанному URL-адресу. При этом можно задать

способ заполнения, позиционирование изображения, а также указать,

должно ли оно перемещаться при прокрутке документа, с помощью

следующих параметров:

― background-image - принимает в качестве значения url (URL-адрес

изображения) или none (отсутствие изображения).

― background-repeat - определяет способ заполнения области

элемента изображением. Возможны следующие значения:

1) repeat - заполнение по горизонтали и по вертикали (по

умолчанию);

2) repeat-x - заполнение по горизонтали;

3) repeat-y - заполнение по вертикали;

4) no-repeat - изображение выводится в единственном экземпляре.

― background-attachment - определяет, будет ли фоновое

изображение прокручиваться при пролистывании документа.

Значением по умолчанию является scroll (будет прокручиваться).

Для фиксации изображения используется значение fixed;

― background-position – определяет начальное положение фонового

изображения с помощью двух значений (горизонтальной и

вертикальной координат), разделенных запятой. Значения

задаются в виде процентов, в абсолютных значениях длины или в

виде ключевых слов: top, center, bottom, left, right. Значения по

умолчанию: left, top;

― background - позволяет установить значения рассмотренных выше

свойств: background-color, background-image, background-repeat и

background-attachment. Эти значения указываются через пробел.

Значение transparent означает отсутствие фона. Пример. Способы

задания фона.

1.3.11. Блочная модель документа. Размеры, поля, отступы, границы

Блочные элементы (блоки текста или box) позволяют оперировать с

текстом в терминах прямоугольников, которые этот текст занимает. При

этом блок текста становится элементом дизайна страницы с теми же

свойствами, что и картинка, таблица или прямоугольная область

приложения. Блок текста обладает свойствами: высоты (height), ширины

(width), границы (border), отступа (margin), набивки (padding ),

произвольного размещения (float), управления обтеканием (clear).

Размеры отступов, границ и полей можно задавать по своему

усмотрению. Однако по умолчанию они имеют значение 0, т. е. не видны и

не занимают место. Параметры height и with определяют размеры

содержимого элемента, отступы задают положение содержимого

относительно рамки, внешние поля позволяют позиционно отделить

содержимое элемента с рамкой от других элементов документа.

Для полей и отступов можно задать размеры 4-х сторон с помощью

следующих параметров:

― margin-top, margin-right, margin-left, margin-bottom – размеры

полей;

― padding-top, padding-right, padding-left, padding-bottom – размеры

отступов.

Значения этих параметров могут быть заданы числом с указанием

единиц измерения или ключевым словом auto.

Кроме того, можно использовать параметры margin и padding с

перечислением через пробел размеров всех четырех сторон в следующем

порядке: top, right, left, bottom. Если указано только одно значение, то оно

будет задавать размер всех сторон. Если указаны два или три значения, то

остальные будут равны размерам соответствующих противоположных

сторон.

Для описания границ блоков применяются следующие атрибуты:

― размеры границы (рамки): border-top-width, border-right-width,

border-left-width, border-bottom-width. Значения можно задавать в

абсолютных единицах (% - нельзя) или ключевые слова: thin

(тонкая), medium (средняя), thick (толстая), по умолчанию -

medium. Также можно использовать параметр border-width с

перечислением размеров всех четырех сторон.

― цвет сторон границы: border-top-color, border-right-color, borderleft-color, border-bottom-color. Параметр border-color можно

использовать для задания сразу всех или только некоторых сторон;

― стиль границы: border-top-style, border-right-style, bor-der-left-style,

border-bottom-style. Может принимать значения: none, dotted,

dashed, solid, double, groove, ridge, inset, outset. Согласно

спецификации CSS1, может быть задан для каждой из границ

блока. Указание типа линии границы поддерживается не всеми

браузерами.

Параметр border-style можно использовать для задания стиля сразу

всех или только некоторых сторон границы. Если стиль границы не задан в

явном виде, другие ее параметры не будут действовать. Задание стиля

(типа) границы в явном виде необходимо, чтобы она была отображена (по

умолчанию граница не видна).

**2. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ**

**2.1. Структура разработанного сайта**

Таблица 1 ­— Структура разработанного сайта

|  |
| --- |
| **Шапка**(header) |
| **Меню**(mainmenu) |
| **Контент**(content) |
| **Подвал**(footer) |

**2.2. Текст программы**

index.html:

<!DOCTYPE HTML>

<html>

<head><meta charset="utf-8">

<title>Автомобильные грузоперевозки</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

<div id="header">

<h2>"Газель"</h2><a href=".html">Заказать обратный звонок</a></div>

<br><div id="mainmenu">

<ul>

<li><a href="#"><div style="background:#00aaff;"><a href="index.html" style="display:block;">Главная</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index2.html" style="display:block;">Тяжелые грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index3.html" style="display:block;">Сборные грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index4.html" style="display:block;">Негабаритные грузы</a></div></a></li>

</ul>

</div>

<div id="content"><h2></h2></div>

Грузоперевозки по Крыму

<p> Мы представляем вам нашу компанию "Газель". В нашей компании достаточно грузоперевозочного транспорта и кваллифицированных водителей для любых ваших запросов. Минимальное время ожидания вам обеспечено.</p>

<p>Делаем грузоперевозки по всему Крыму быстро и надежно. График работы: ежедневно круглосуточно.</p>

<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><div id="footer"><h5>г. Севастополь<br>

Менеджер проекта: +79780000000</br>

Отдел аренды: +79781111111</h5></div>

</body>

</html>

index2.html:

<!DOCTYPE HTML>

<html>

<head><meta charset="utf-8">

<title>Автомобильные грузоперевозки</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

<div id="header">

<h2>"Газель"</h2><a href=".html">Заказать обратный звонок</a></div>

<br><div id="mainmenu">

<ul>

<li><a href="#"><div style="background:#00aaff;"><a href="index.html" style="display:block;">Главная</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index2.html" style="display:block;">Тяжелые грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index3.html" style="display:block;">Сборные грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index4.html" style="display:block;">Негабаритные грузы</a></div></a></li>

</ul>

</div>

<div id="content"><h2></h2></div>

<strong>Тяжелые грузы</strong>

<p><img src="i2.PNG" width="400" height="300"></p>

<p>Доставим без проблем за промежуток времени от недели до 1 месяца.</p>

<p>Цену узнать позвонив нам.</p>

<br><br><br><br><br><div id="footer"><h5>г. Севастополь<br>

Менеджер проекта: +79780000000</br>

Отдел аренды: +79781111111</h5></div>

</body>

</html>

index3.html:

<!DOCTYPE HTML>

<html>

<head><meta charset="utf-8">

<title>Автомобильные грузоперевозки</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

<div id="header">

<h2>"Газель"</h2><a href=".html">Заказать обратный звонок</a></div>

<br><div id="mainmenu">

<ul>

<li><a href="#"><div style="background:#00aaff;"><a href="index.html" style="display:block;">Главная</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index2.html" style="display:block;">Тяжелые грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index3.html" style="display:block;">Сборные грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index4.html" style="display:block;">Негабаритные грузы</a></div></a></li>

</ul>

</div>

<div id="content"><h2></h2></div>

<strong>Сборные грузы</strong>

<p><img src="i.png" width="400" height="300"></p>

<p>Доставим без проблем за промежуток времени от недели до 2 недель.</p>

<p>Цену узнать позвонив нам.</p>

<br><br><br><br><br><div id="footer"><h5>г. Севастополь<br>

Менеджер проекта: +79780000000</br>

Отдел аренды: +79781111111</h5></div>

</body>

</html>

index4.html:

<!DOCTYPE HTML>

<html>

<head><meta charset="utf-8">

<title>Автомобильные грузоперевозки</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

<div id="header">

<h2>"Газель"</h2><a href=".html">Заказать обратный звонок</a></div>

<br><div id="mainmenu">

<ul>

<li><a href="#"><div style="background:#00aaff;"><a href="index.html" style="display:block;">Главная</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index2.html" style="display:block;">Тяжелые грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index3.html" style="display:block;">Сборные грузы</a></div></a></li>

<li><a href="#"><div style="background:#00aaff;"><a href="index4.html" style="display:block;">Негабаритные грузы</a></div></a></li>

</ul>

</div>

<div id="content"><h2></h2></div>

<strong>Негабаритные грузы</strong>

<p><img src="s1200.png" width="400" height="300"></p>

<p>Доставим без проблем за промежуток времени от 3 дней до 2 недель.</p>

<p>Цену узнать позвонив нам.</p>

<br><br><br><br><br><div id="footer"><h5>г. Севастополь<br>

Менеджер проекта: +79780000000</br>

Отдел аренды: +79781111111</h5></div>

</body>

</html>

style.css:

#header {

background: #00CCCC;

width: 100%;

height: 80px;

text-align: center;

margin-top: -20px;

}

#mainmenu {

background: #00aaff;

position: relative;

overflow: hidden;

height: 40px;

margin: 30px 0;

margin-top: -19px;

}

#mainmenu ul {

list-style: none;

margin: 0;

padding: 0;

position: relative;

left: 50%;

float: left;

font: 14px Arial, Helvetica, sans-serif;

height: 40px;

}

#mainmenu ul li {

position: relative;

left: -50%;

float: left;

margin: 0 10px;

height: 40px;

}

#mainmenu ul li a {

color: black;

display: block;

text-decoration: none;

padding: 0 15px;

line-height: 40px;

}

#mainmenu ul li a:hover {

background-color: #0066FF;

}

#content {

width: 100%;

height: 20px;

font-family: Arial, sans-serif;

font-size: 14px;

text-align: left;

}

#footer {

bottom: 0;

width: 100%;

height: 100%;

background-color: #0066FF;

text-align: center;

}

**2.3. Таблица свойств CSS стилей использованные для создания верстки документа**

Таблица 2.2— свойства CSS стилей использованные для создания верстки документа

|  |  |
| --- | --- |
| Имя свойства | Назначения свойства |
| background | Короткое свойство, устанавливающее все свойства фона элемента за одну декларацию |
| width | Задает ширину элемента |
| height | Устанавливает высоту блочных или заменяемых элементов |
| text-align | Устанавливает горизонтальное выравнивание текста элемента |
| margin-top | Определяет размер свободного поля (верхнего) |
| position | Определяет размещение элемента относительно других элементов |
| overflow | Управляет отображением содержимого элемента за пределами области вывода |
| list-style | Задает в сокращенном виде тип списка, расположение маркеров и их изображение |
| padding | Задает расстояние между всеми границами элемента и его содержимым |
| float | Задает обтекание элемента текстом |
| font | Универсальное свойство, позволяющее определить все свойства шрифта за одну декларацию |
| font-family | Задает семейство шрифта, которое будет использоваться для оформления текста содержимого |

**2.3. Скриншоты выполнения программы**

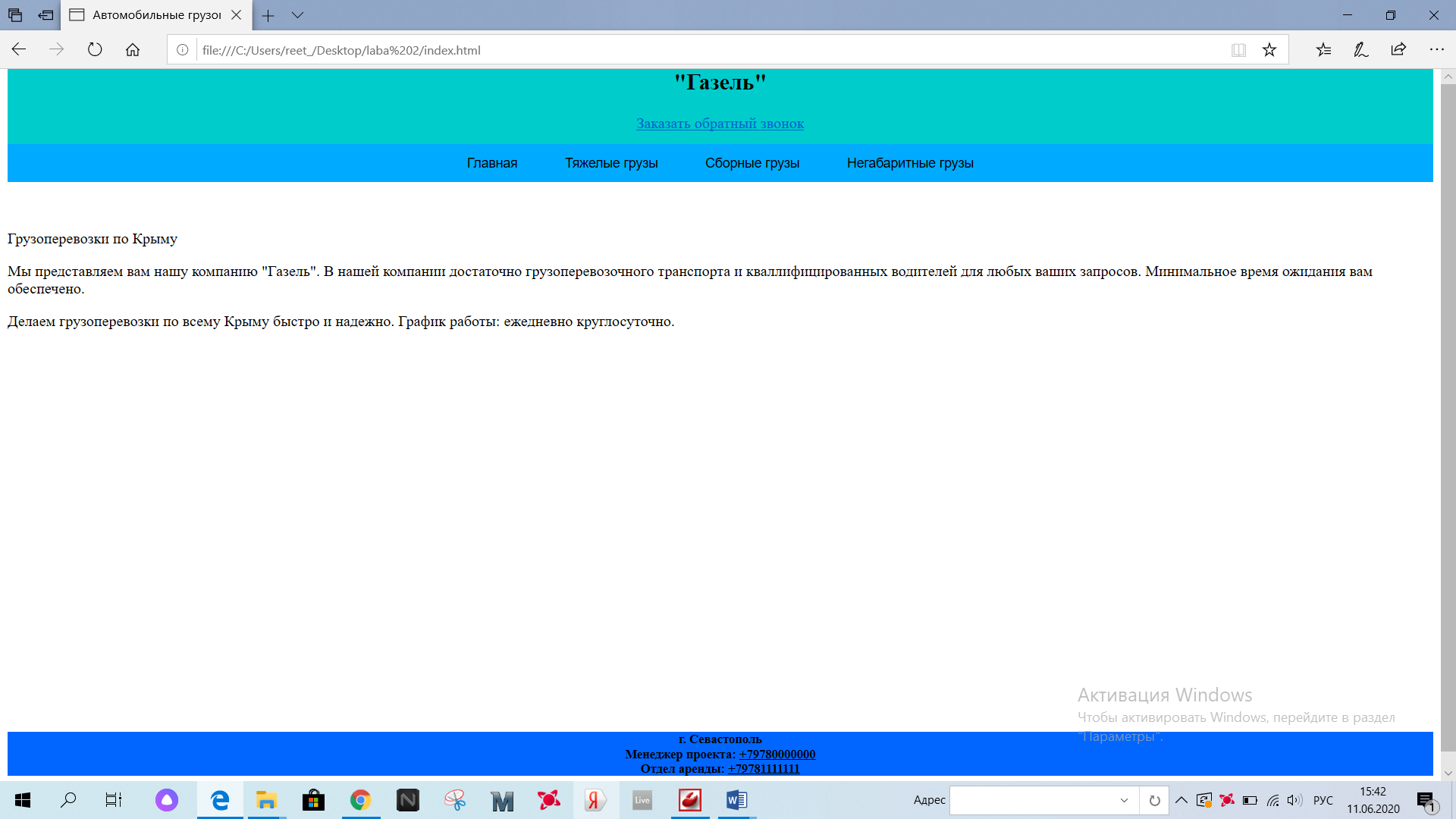


Рис.1.1 — скриншот выполнения программы первой страницы документа

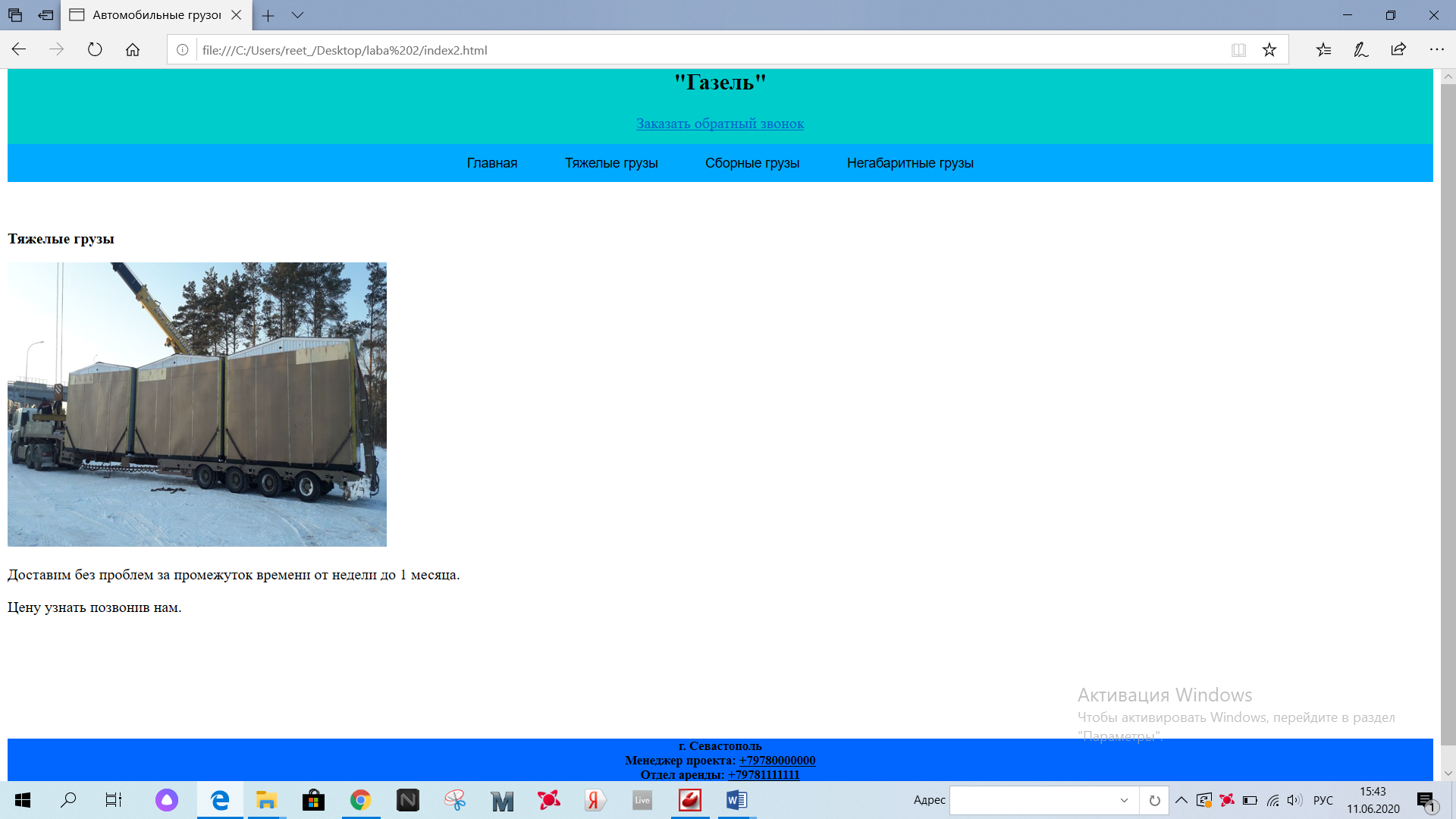


Рис. 1.2 — скриншот выполнения программы второй страницы документа

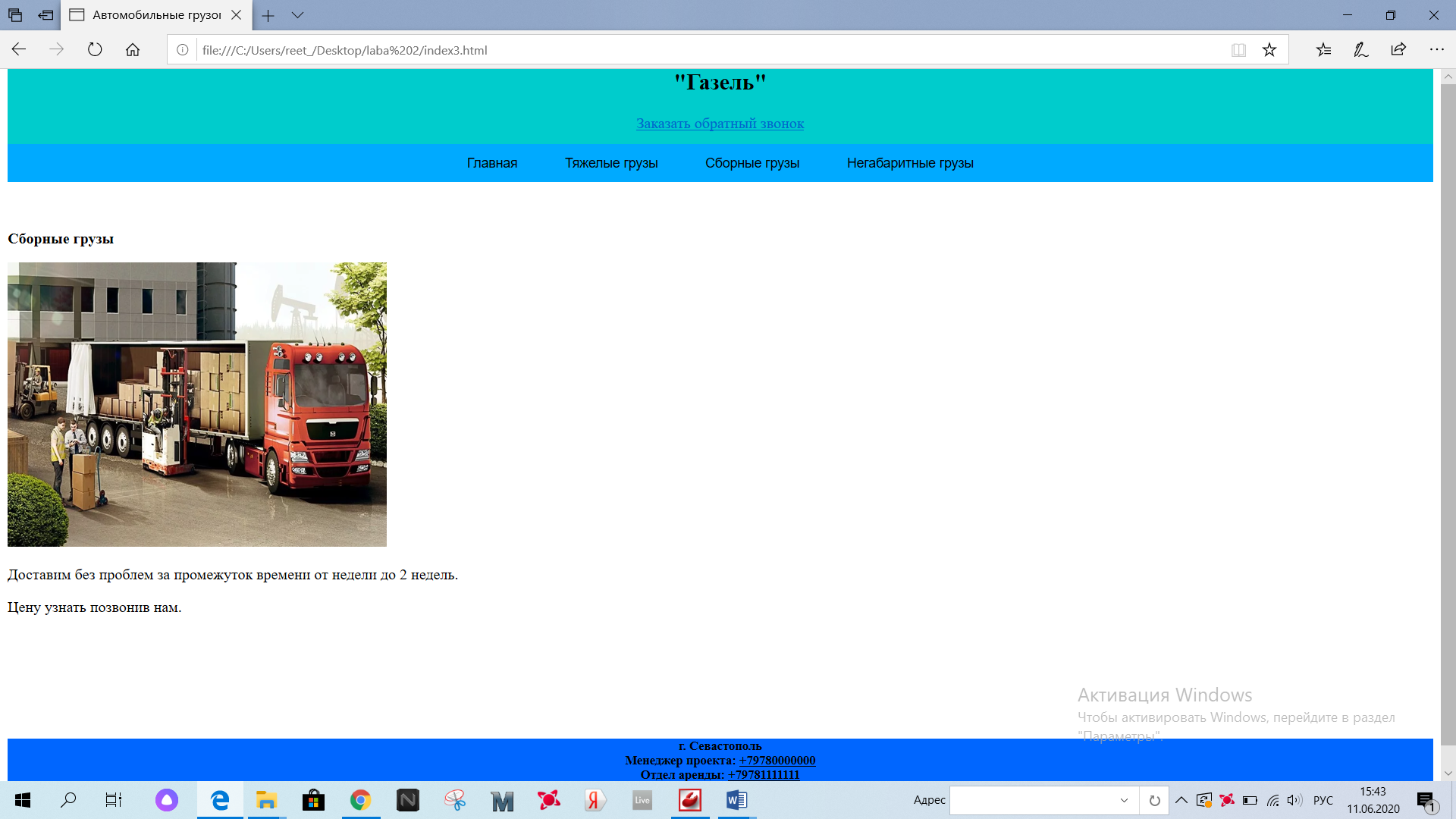


Рис. 1.3 — скриншот выполнения программы третей страницы документа

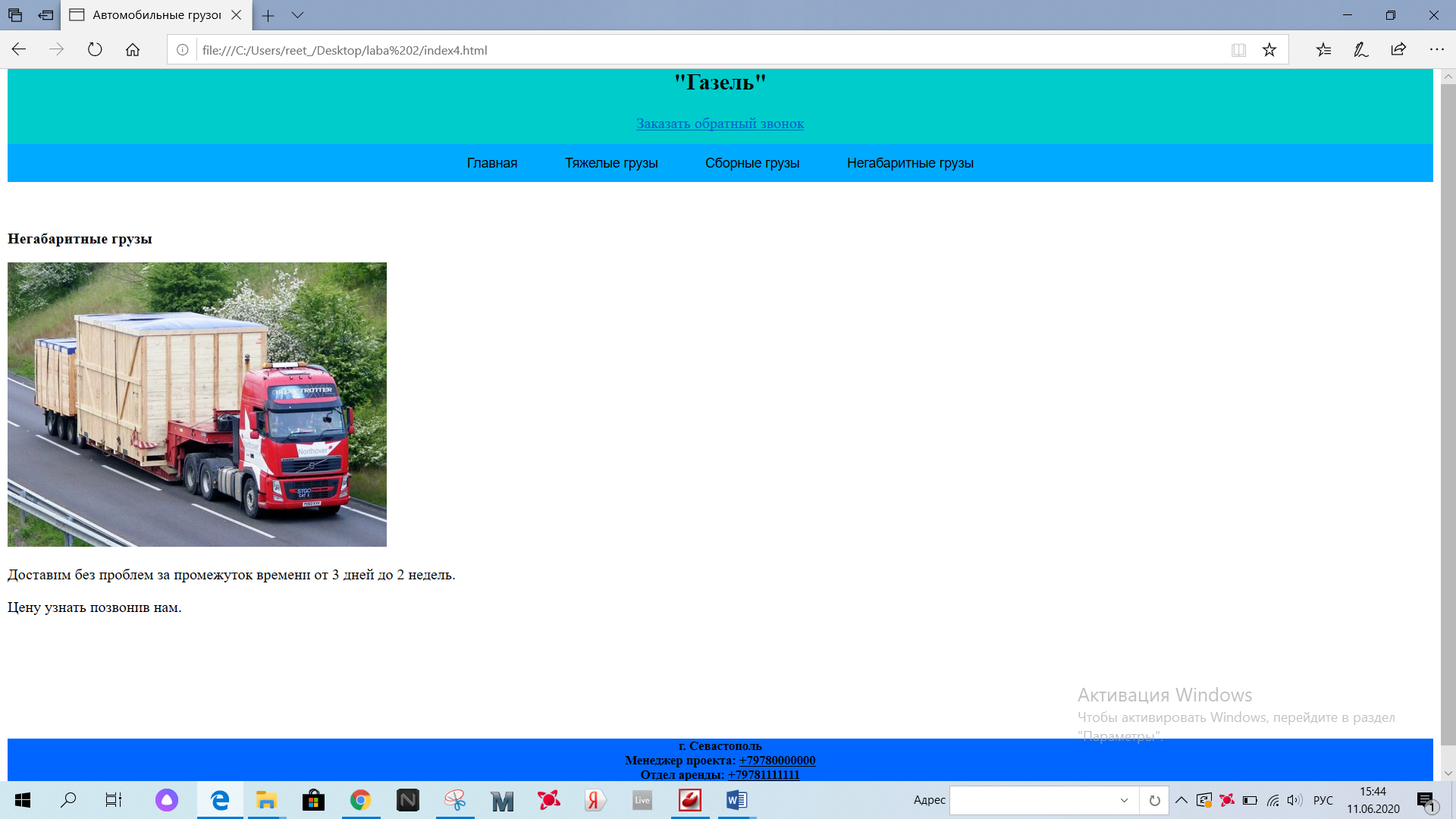


Рис. 1.4 — скриншот выполнения программы четвертой страницы документа

**ВЫВОДЫ**

Задачей, выполненной работы, было научиться работать с каскадными таблицами стилей, а также изучить основные методы работы с селекторами и построения сайта блочной верстки. Создал сайт, состоящий и нескольких веб -страниц согласно варианту. К плюсам программы CodeLobster можно отнести несложные манипуляции при создании документа и работы с кодом.