**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Ивановский государственный энергетический университет имени**

**В.И. Ленина» (ИГЭУ)**

**Отчет по лабораторной работе №3**

**3D-Графика(HTML,CSS,JS)**

|  |
| --- |
| **Вариант №10** |
| **Выполнила: студентка гр.2-80к** |
| **Данченко Е.А.** |
| **Принял: Зубков В.П.** |
| **Дата приема:** |

**Иваново 2024**

**Задание 10.** Используя HTML, javascript, CSS3, нарисовать на экране четырехугольную 3D-пирамиду, расположенную под углом к горизонту. Параметры фигуры в пикселах и координаты верхнего левого угла прямоугольной области изображения вводить в диалоге.

**Анализ задачи**

Для решения данной задачи понадобятся:

1. **Структура HTML** (подробно описана в Лаб 1.docx и Лаб 2.docx)
2. в коде используются теги HTML, которые определяют структуру и содержание веб-страницы.
3. **CSS стили** CSS-файл "Лаб3.css", включен с помощью тега <link> в html-файл, содержит стили для пирамиды.
4. С**електоры CSS** ( .pyramid, .side., base, .one, .two, .three, .four ) для выбора элементов, которые собираемся стилизовать.
5. **Свойства CSS** **и их значения используемые в коде**:

* position: relative - устанавливает элементу относительное позиционирование, то есть он будет располагаться в соответствии с его исходной позицией, но при этом его местоположение может быть изменено с помощью свойств top, right, bottom и left.
* display: flex - устанавливает элементу блочное отображение с возможностью гибкой настройки расположения и размеров его дочерних элементов
* min-height: 100vh - устанавливает минимальную высоту элемента в 100% высоты окна браузера
* transform-style: preserve-3d - определяет, как элементы внутри данного элемента будут позиционироваться в трехмерном пространстве
* transform: rotateY(30deg) rotateX(-30deg) rotateZ(30deg) - задает элементу преобразование, в данном случае вращение по осям Y, X и Z на указанные градусы
* position: absolute - устанавливает элементу абсолютное позиционирование, то есть он будет располагаться относительно ближайшего позиционированного родителя или окна браузера, если такого нет
* top: 400px - устанавливает верхнюю границу элемента на расстоянии 400 пикселей от его родителя или окна браузера
* left: 400px - устанавливает левую границу элемента на расстоянии 400 пикселей от его родителя или окна браузера
* width: 200px - устанавливает ширину элемента на 200 пикселей
* height: 200px - устанавливает высоту элемента на 200 пикселей
* border: 100px solid transparent - устанавливает границу элемента толщиной 100 пикселей, цвет границы не указан, поэтому она будет прозрачной
* border-bottom: 200px solid red - устанавливает нижнюю границу элемента толщиной 200 пикселей и красного цвета
* border-top: 0px - устанавливает верхнюю границу элемента толщиной 0 пикселей, то есть ее нет
* transform: translateY(73px) rotateX(90deg) - задает элементу преобразование, в данном случае сдвиг по оси Y на 73 пикселя и вращение по оси X на 90 градусов
* border: 0 - устанавливает границу элемента толщиной 0 пикселей, то есть ее нет
* background: blue - устанавливает фон элемента синего цвета
* transform: rotateX(30deg) - задает элементу преобразование, в данном случае вращение по оси X на 30 градусов
* transform-origin: 0 0 - устанавливает точку, относительно которой будет происходить преобразование элемента
* transform-origin: 100px 0 - устанавливает точку, относительно которой будет происходить преобразование элемента, в данном случае это 100 пикселей от левой границы элемента и 0 пикселей от верхней границы
* transform: rotateY(90deg) rotateX(-30deg) - задает элементу преобразование, в данном случае вращение по оси Y на 90 градусов и вращение по оси X на -30 градусов (то есть в обратном направлении)
* border-bottom-color: purple - устанавливает цвет нижней границы элемента в фиолетовый

1. **JavaScript**

Код динамически применяет стили к элементу пирамидных слоев на основе вводимых пользователем данных в диалоговых окнах:

**Функция prompt()** для сбора пользовательского ввода для поворотов и координат. Пользователю предлагается указать углы поворота (X, Y, Z) – параметры пирамиды и координаты (X, Y) верхнего левого угла интересующей прямоугольной области.

1. **DOM**

Код JavaScript взаимодействует с HTML-документом с помощью методов DOM (**Document Object Model**). Он извлекает элементы с помощью свойства **document.querySelector()** и изменяет их стили (**style**).

**Код задачи**

<!DOCTYPE html>

<html **lang**=**"ru"**>

<head>

<meta **charset**=**"UTF-8"**>

<title>3D Pyramid</title>

<link **rel**=**"stylesheet"** **href**=**"Лаб3.css"**>

</head>

<body>

<div **class**=**"pyramid"**>

<div **class**=**"base"**></div>

<div **class**=**"side one"**></div>

<div **class**=**"side two"**></div>

<div **class**=**"side three"**></div>

<div **class**=**"side four"**></div>

</div>

<script>

***var*** x\_rotate **=** parseInt**(**prompt**(**"Укажите поворот пирамиды по оси X (в градусах):"**));**

***var*** y\_rotate **=** parseInt**(**prompt**(**"Укажите поворот пирамиды по оси Y (в градусах):"**));**

***var*** z\_rotate **=** parseInt**(**prompt**(**"Укажите поворот пирамиды по оси Z (в градусах):"**));**

***var*** x **=** parseInt**(**prompt**(**"Введите координату X верхнего левого угла прямоугольной области изображения:"**));**

***var*** y **=** parseInt**(**prompt**(**"Введите координату Y верхнего левого угла прямоугольной области изображения:"**));**

***let*** pyramid **=** document.querySelector**(**'.pyramid'**);**

pyramid.style.transform **=** "rotateY(" **+** y\_rotate **+** "deg) rotateX("**+** x\_rotate **+** "deg) rotateZ(" **+** z\_rotate **+** "deg)"**;**

pyramid.style.left **=** x **+** 'px'**;**

pyramid.style.top **=** y **+** 'px'**;**

</script>

</body>

</html>

Данный код прилагается в файле Лаб3.html.

body **{**

**position: relative;**

**display: flex;**

**min-height: 100vh;**

**}**

**.**pyramid **{**

**position:absolute ;**

**top: 400px;**

**left: 400px;**

**width: 200px;**

**height: 200px;**

**transform-style: preserve-3d;**

**transform: rotateY(30deg) rotateX(-30deg) rotateZ(30deg);**

**}**

**.**side **{**

**width: 0;**

**height: 0;**

**position: absolute;**

**opacity: 0.7;**

**border: 100px solid transparent;**

**border-bottom: 200px solid red;**

**border-top: 0px;**

**}**

**.**base **{**

**position: absolute;**

**width: 100%;**

**height: 100%;**

**transform: translateY(73px) rotateX(90deg);**

**border: 0;**

**background: blue;**

**}**

**.**one **{**

**transform: rotateX(30deg);**

**transform-origin: 0 0;**

**}**

**.**two **{**

**transform-origin: 100px 0;**

**transform: rotateY(90deg) rotateX(-30deg);**

**border-bottom-color: purple;**

**}**

**.**three **{**

**transform-origin: 100px 0;**

**transform: rotateY(270deg) rotateX(-30deg);**

**border-bottom-color: hotpink;**

**}**

**.**four **{**

**transform-origin: 100px 0;**

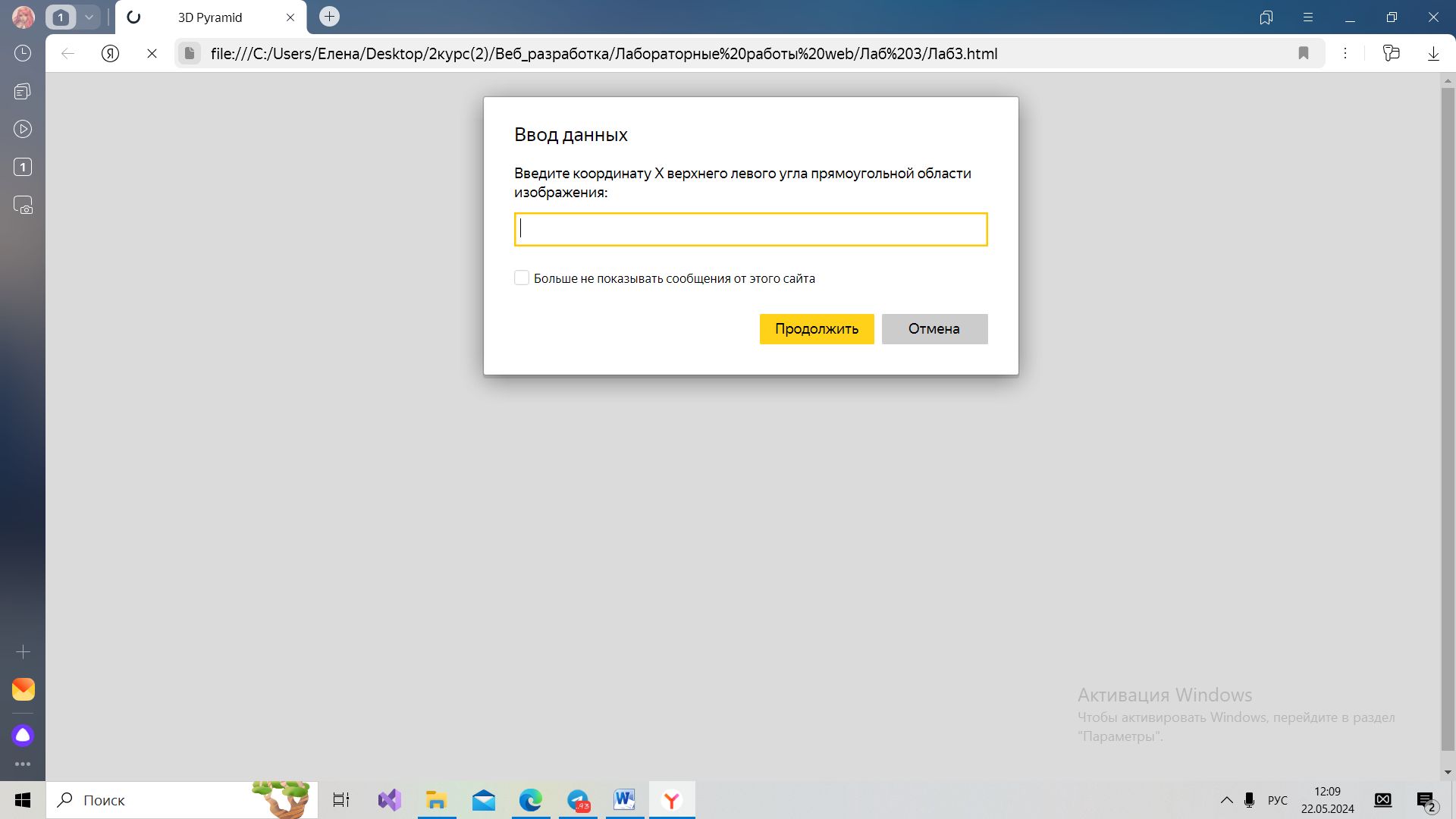
**transform: rotateY(0) rotateX(-30deg);**

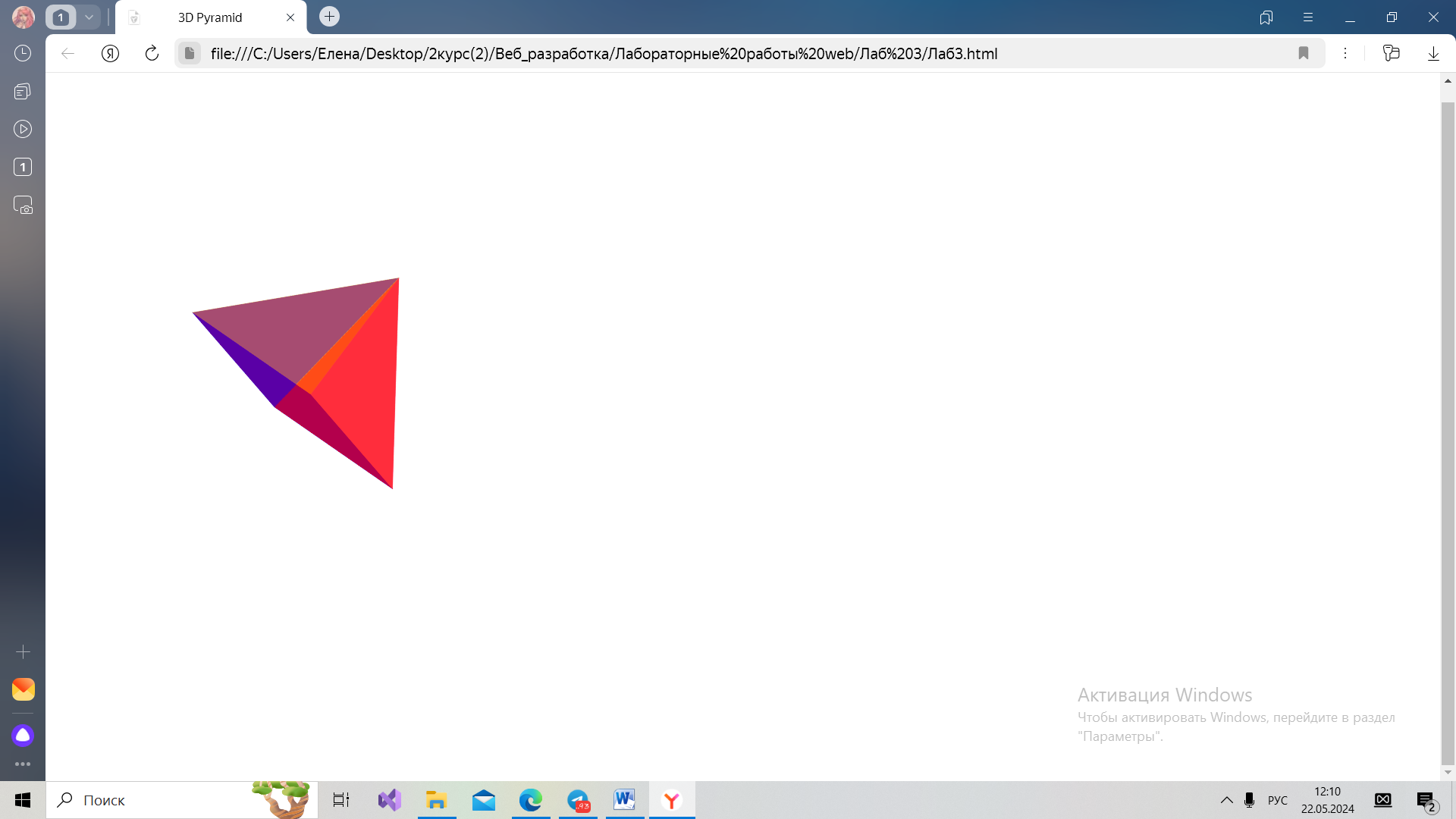
**border-bottom-color: yellow;**

**}**

Данный код прилагается в файле Лаб3.css

**Скриншоты результат работы кода**





**Заключение**

В ходе выполнения лабораторной работы по теме "**3D-Графика** " были изучены: html-разметка страницы, использование различных свойств CSS для кастомизации внешнего вида фигуры, эелементы JS и DOM для взаимодействия с пользователем. Полученные знания и навыки будут полезны при создании веб-страниц с 3D - фигурами. Данная лабораторная работа показывает, как технологии (HTML, CSS, JS) могут работать вместе для создания интерактивных и динамических веб-страниц.