**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря** **Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота №2**

з дисципліни «Основи Front-end технологій»

Виконав: Перевірив:

Студент групи ІМ-13 Жереб К.А.

Яцук Владислав Вікторович

Київ 2023

Тема: «Мова гіпертекстової розмітки сайту HTML-5. Таблиці. Каскадна таблиця стилів CSS-3»

**Завдання 1.**

За допомогою мови HTML-5 створити таблицю відповідно за своїм варіантом. У перший рядок таблиці розташувати ПІБ виконавця (окремо в кожну комірку). У другому рядку записати послідовно групу, номер лабораторної роботи та номер варіанту.

В цьому ж файлі додати CSS властивості:

Для 1-го рядка встановити колір #nn8f8f та 2 #f5f5nn (властивість background-color), де nn- номер вашого варіанту.

Для тексту обрати контрастний колір за допомогою сервісу [color.adobe.com](https://color.adobe.com/)

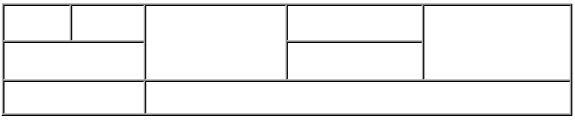
Зробити таким чином, щоб при наведенні курсора на 1-й та 2-й рядки кольори змінювалися на протилежні (використати псевдо-клас hover).

**Завдання2.**

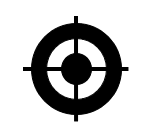
Створити зображення відповідно до свого варіанта: 1. – за допомогою position; 2- за допомогою transform , використовуючи тільки властивості CSS. Властивості CSS зберегти в окремому файлі style.css.

Мій варіант: 4 (54 % 10 = 4)

Таблиця, яку потрібно створити:



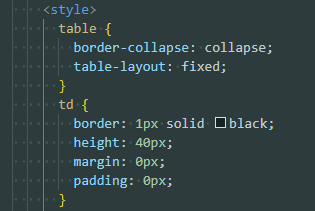
Фігура, яку потрібно створити:

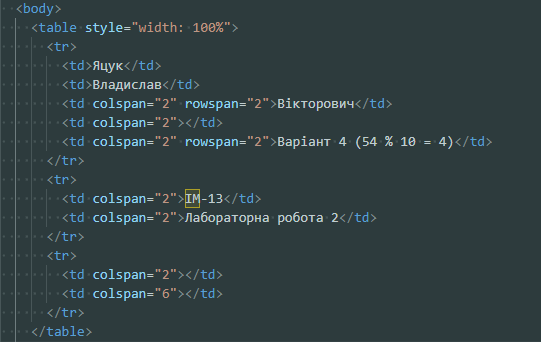


Виконання завдання:

Завдання 1

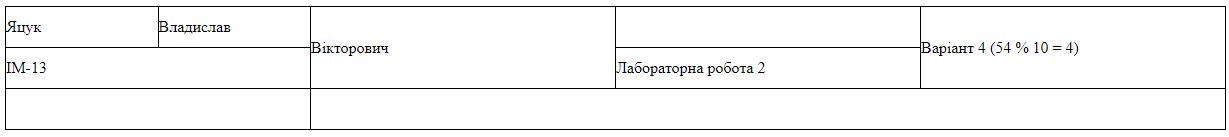
Створення таблиці:



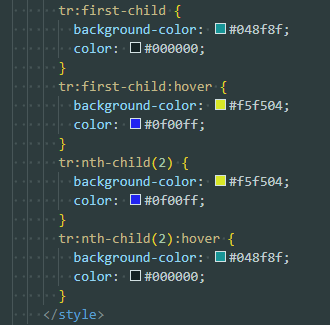


Атрибути rowspan та colspan допомогли об’єднати декілька клітинок таблиці одного рядка/стовпця в одну.

Результат:



Додаємо кольори до таблиці:

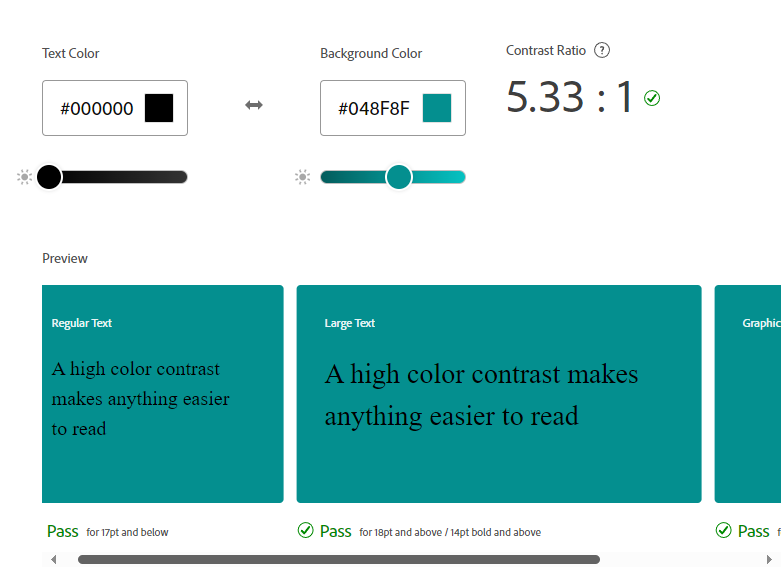


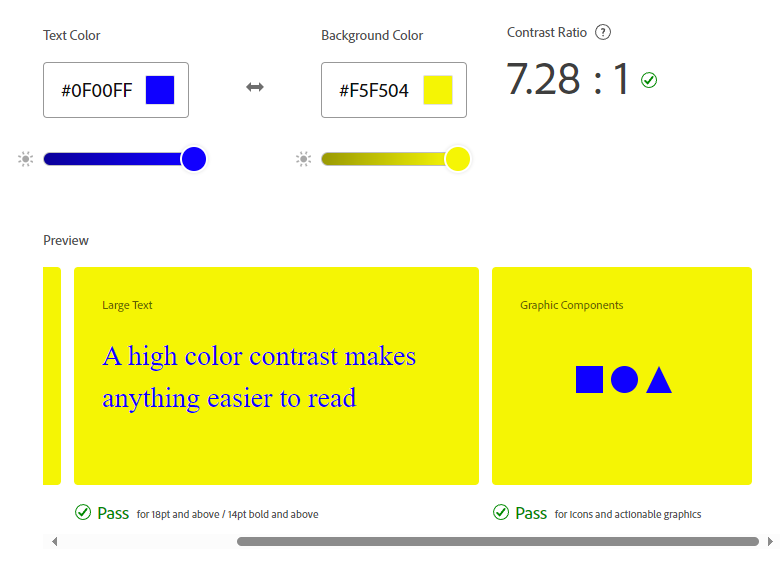
Псевдоклас :first-child допоміг зафарбувати саме перший рядок таблиці (можна було використати nth-child(1)), nth-child(2) – зафарбувати саме другий рядок таблиці, а :hover – змінювати колір фону і тексту при наведенні на відповідні рядки таблиці курсором.

Результат:



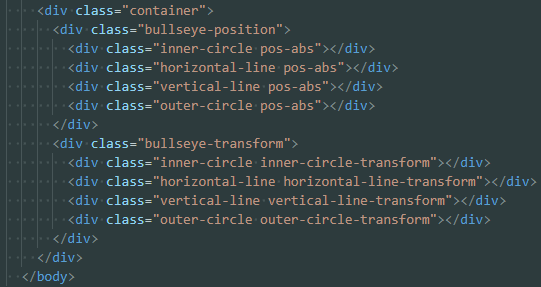
Кольори мають хороший contrast ratio:





Завдання 2

Зміна структури html файлу:



Зовнішній div буде використаний для того, щоб помістити обидві фігури в один ряд (використання display: flex). Відповідно перший внутрішній div буде містити фігуру, створену з використанням position, а другий – з використанням transform.

Фігуру можна поділити на елементарні фігури (внутрішнє й зовнішнє кола, вертикальна й горизонтальна лінії), для їх створення я створив відповідні класи – inner-circle, outer-circle, horizontal-line, vertical-line.

Вміст файлу style.css:

.container {

  display: flex;

  padding-top: 10px;

}

.bullseye-position {

  width: 175px;

  height: 175px;

  position: relative;

  display: flex;

  justify-content: center;

  align-items: center;

  margin-right: 10px;

}

.inner-circle {

  width: 50px;

  height: 50px;

  background-color: black;

  border-radius: 50%;

}

.pos-abs {

  position: absolute;

}

.horizontal-line {

  width: 175px;

  height: 5px;

  background-color: black;

}

.vertical-line {

  width: 5px;

  height: 175px;

  background-color: black;

}

.outer-circle {

  width: 150px;

  height: 150px;

  border: 25px solid black;

  border-radius: 50%;

  box-sizing: border-box;

}

.bullseye-transform {

  width: 175px;

  height: 175px;

}

.inner-circle-transform {

  transform: translate(calc((175px - 50px) / 2), calc((175px - 50px) / 2));

}

.horizontal-line-transform {

  transform: translate(0px, calc(-50px + (175px - 5px) / 2));

}

.vertical-line-transform {

  transform: translate(calc((175px - 5px) / 2), calc(-50px + -5px));

}

.outer-circle-transform {

  transform: translate(

    calc((175px - 150px) / 2),

    calc((-50px + -5px + -175px) + (175px - 150px) / 2)

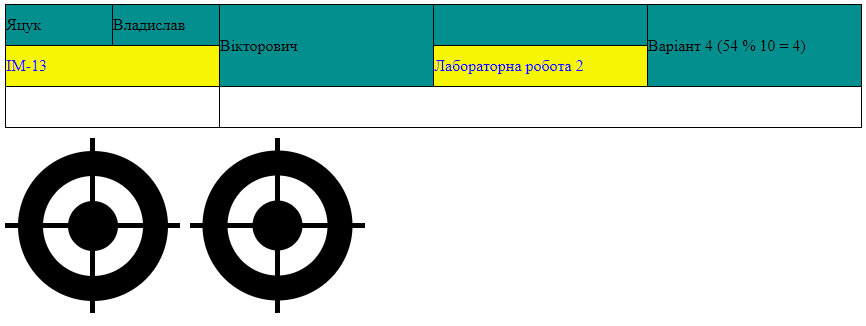
  );

}

Оскільки фігура симетрична, в першому випадку я вирішив використати positioin: absolute для кожного елементу фігури, та display:flex для контейнеру, щоб потім відцентрувати кожен елемент з допомогою justify-content: center і align-items: center.

В другому ж випадку я використовував функції translate() і calc() для обчислення, на скільки потрібно перемістити кожен елемент фігури. Для цієї фігури цей спосіб виявився набагато менш зручним.

Результат:



Висновок: за час виконання лабораторної роботи я навчився створювати таблиці з допомогою HTML-5, використав на практиці базові можливості CSS-3 для стилізації веб-сторінки, такі як селектори класів, псевдокласи (наприклад, hover для розширення взаємодії між користувачем і веб-сторінкою), поверхнево дізнався про розміщення (layout) елементів на веб-сторінці, що є дуже важливим концептом для розуміння для створення хороших веб-сторінок.

Посилання:  
[GitHub Pages](https://vladyatsuk.github.io/Frontend_labs/Lab_2/index.html)

[Репозиторій](https://github.com/vladyatsuk/Frontend_labs)