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 №5**

з дисципліни «Основи Front-end технологій» на тему

«JavaScript. Події. Регулярні вирази»

Виконав: Перевірив:

студент II курсу ФІОТ Жереб К.А.

групи ІМ-32 Лаврик Севастьян Миколайович  
номер у списку групи: 54

Київ 2025

Завдання1.

Вибрати рядок у таблиці 1, номер якого збігається з Вашим варіантом. Для формування форми взяти рядки, які відповідають Вашому варіанту таблиці2.

З отриманої інформації сформувати форму. За допомогою регулярних виразів JS виконати перевірку валідності введеної інформації. Якщо все правильно – вивести введену інформацію в окремому вікні, якщо є помилки – виділити рядки, які містять помилку.

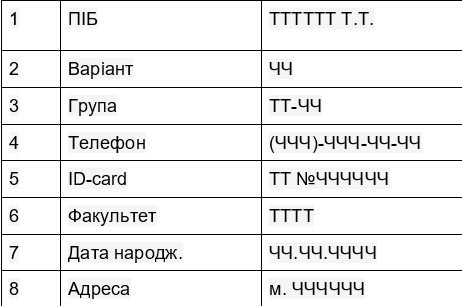
Завдання2.

Створити таблицю розміром (6х6). Клітинки таблиці заповнюються послідовно номерами від 1 до 36 по рядках. При наведенні на клітинку, що відповідає номеру варіанта виконується зміна кольору на випадковий, при Click на ній - зміна кольору на обраний з палітри, а при dblClick:

Варіант 54 ( ≈4):



Форма буде складатися з інформації з рядів 1, 5, 6, 7, 8, які є:



ПІБ, ID-Card, Факультет, Дата народж., Адреса.

В другому завданні при варіанті 4 dblClick виконує «зміна кольору клітинок побічної діагоналі таблиці».

Хід роботи:

Для виконання завдань я створив два окремих div блоки для 1 і 2 завдання. В першому блоці я створив просту форму за допомогою тегів <form> та <input> з типом text окрім дати народження, де тип встановлений date. Після приукрашення форми я приступив до імплементування функціоналу форми.

Коли користувач вводить усі дані та натискає кнопку submit, інформація передається як параметр до функції formValidation, де вона форматується в словник, для більш простого подальшого процесу валідації. Далі словник передається у функцію validateInfo, де записані регулярні виразі для кожної ячейки даних у формі. Потім ці регулярні вирази порівнюються з інформацією у словниках, і якщо усі дані пройшли верифікацію, то флаг isValid залишається true, і функція передає словник у наступню: outputResult, в якій вже відбувається додавання інформації на сторінку html. Також варто додати, що якщо один або декілька елементів не проходять верифікацію, флаг isValid стає false та до цих елементів додається клас invalid, в якому описаний стиль тега input при неправильно введених даних. При повторній перевірці елементів, якщо верифікація пройдена, до клас прибирається та флаг isValid знову стає true.

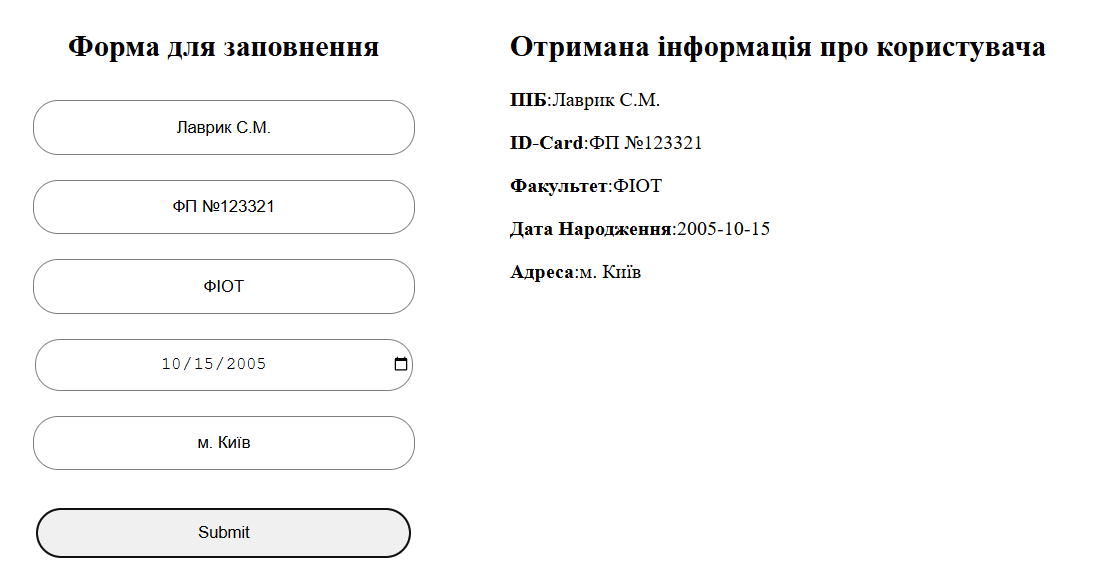
Для другого завдання я спочатку створив таблицю 6х6, де до клітинки з цифрою 4 я додав 3 атрибути onmouseover, onclick та ondblclick, в які я додав відповідні функції з js.

Для onmouseover я додав функцію randomRGB, яка створює три рандомних числа від 0 до 255, і надає цьому елементу властивість backgroundColor: ${згенероване rgb}

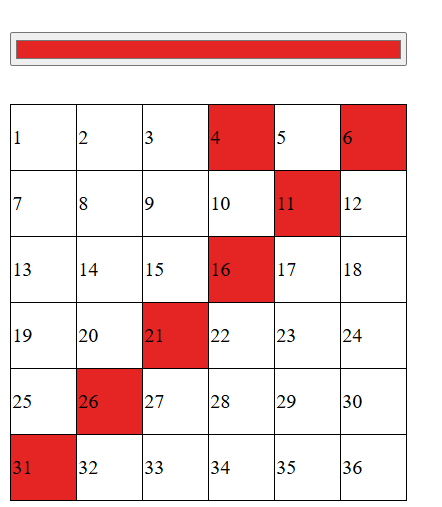
Для onclick я додав функцію ChangeColorOnClick, яка бере поточне значення тегу <input type=”color”>, та присваює це значення властивості backgroundColor цього елементу. В теорії цю функцію можна було повністю вмістити в атрибут onclick у файлі html, так як функція однолінійна, але вона занадто довга, і виглядало б воно не дуже гарно, тому я не став цього робити.

Для ondblclick я додав функцію changeColorOnDblClick, який працює за схожим принципом до функції з onclick, але в ньому присутній флаг isColored, за допомогою якого функція може як додавати колір до елементів діагоналі, так і прибирати з них колір. При подвійному натисканні функція збирає усі елементи з класом .dblClick та за допомогою метода forEach дає кожному елементу значення з того самого ж тега <input>. А якщо флаг isColored є true, то функція просто змінює значення backgroundColor на ””.

Отриманий результат:







Посилання на проєкт Github:

<https://github.com/LeVasTiaN/Frontend/tree/main/Lab5>

Посилання на створений сайт:

<https://levastian.github.io/Frontend/Lab5/Lab5.html>