**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**

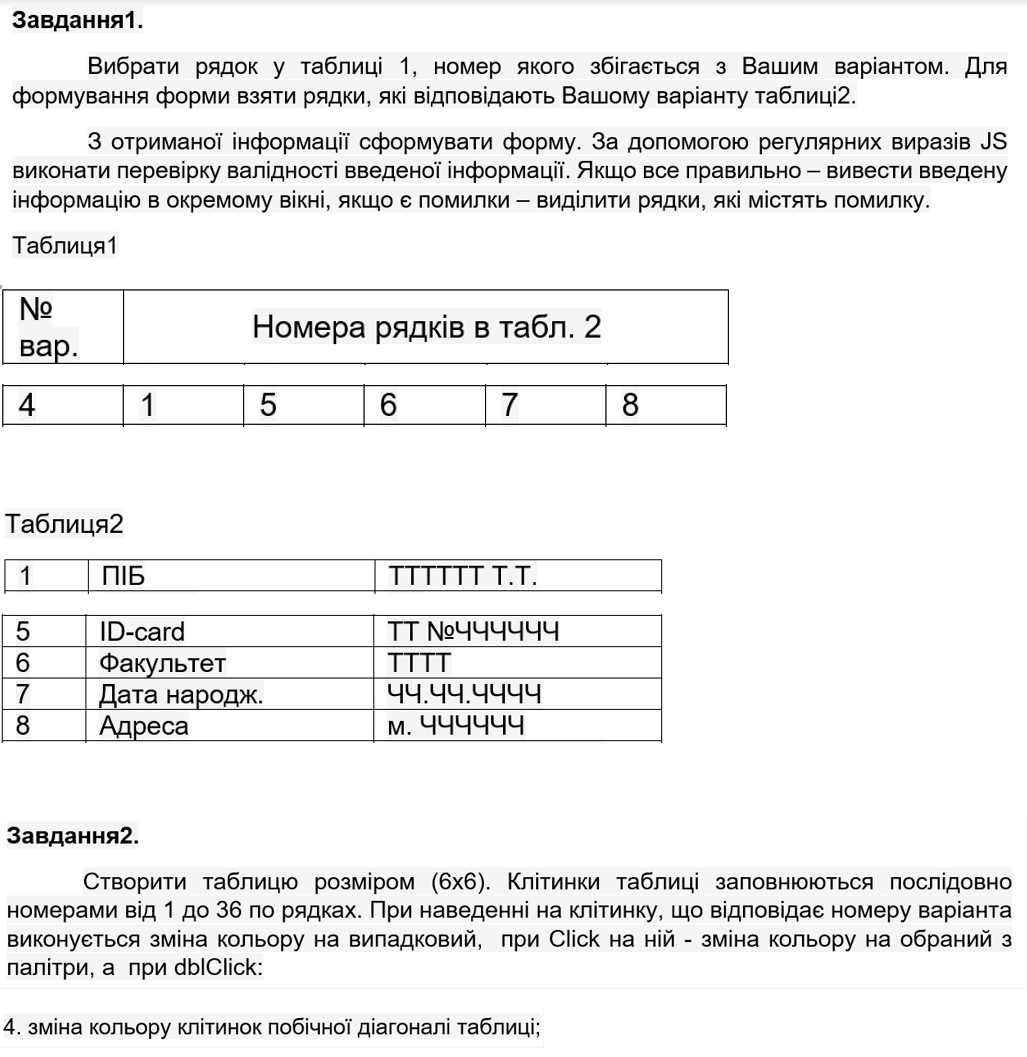
з дисципліни  
«Основи Front-end технологій»

Виконав: Перевірив:

студент групи ІМ-11 [Костянтин Анатолійович Жереб](https://fictadvisor.com/teachers/633ed74e-87ca-4c57-bceb-b52b5cb2249c)  
Царик Микола Миколайович  
варіант відповідно до списку: 34

**Київ 2023**

**Завдання за варіантом**

****

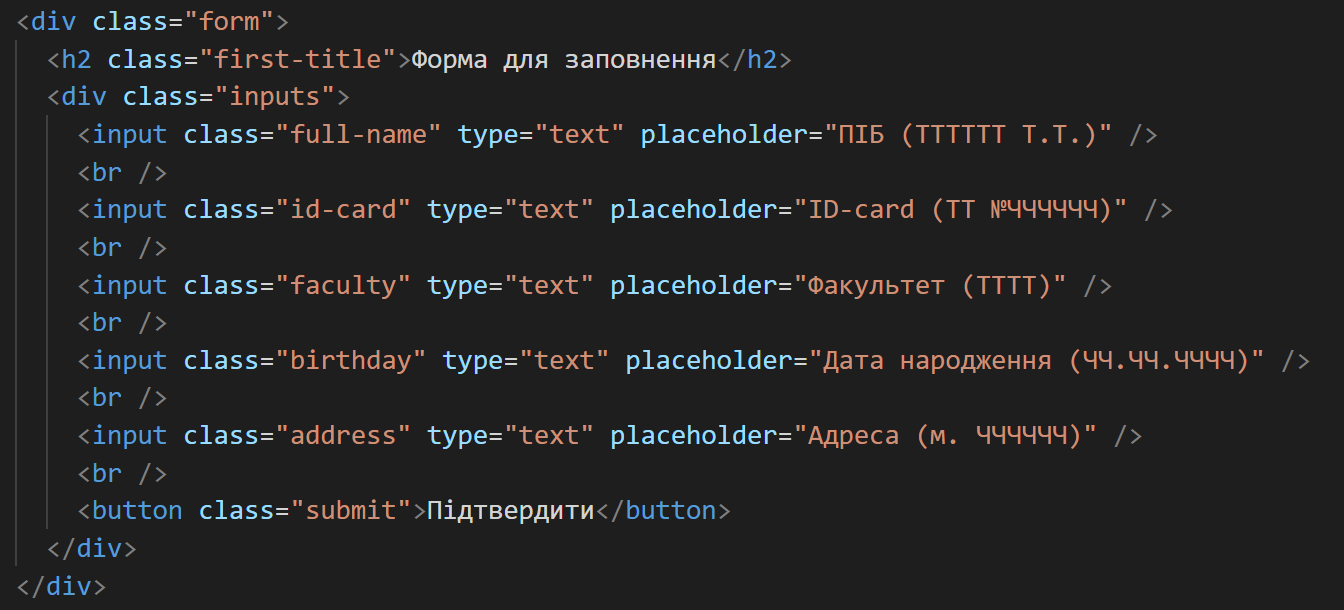
Мій варіант: **34**

Варіант для першого завдання: (34 mod 10) = **4**

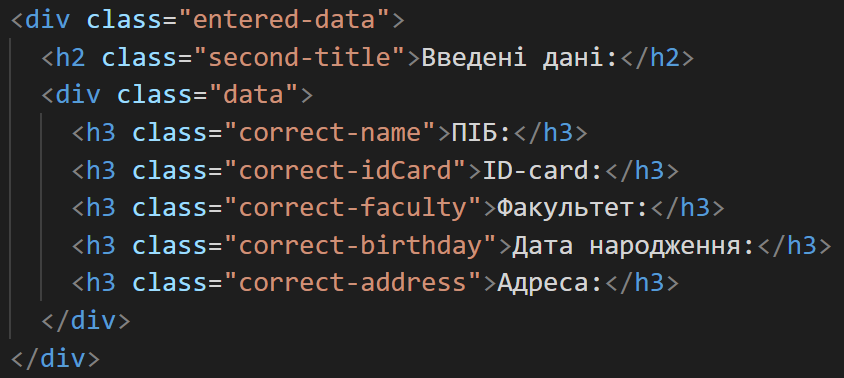
**Хід роботи**

**Завдання 1**

1) Створимо базову стуктуру до нашої форми

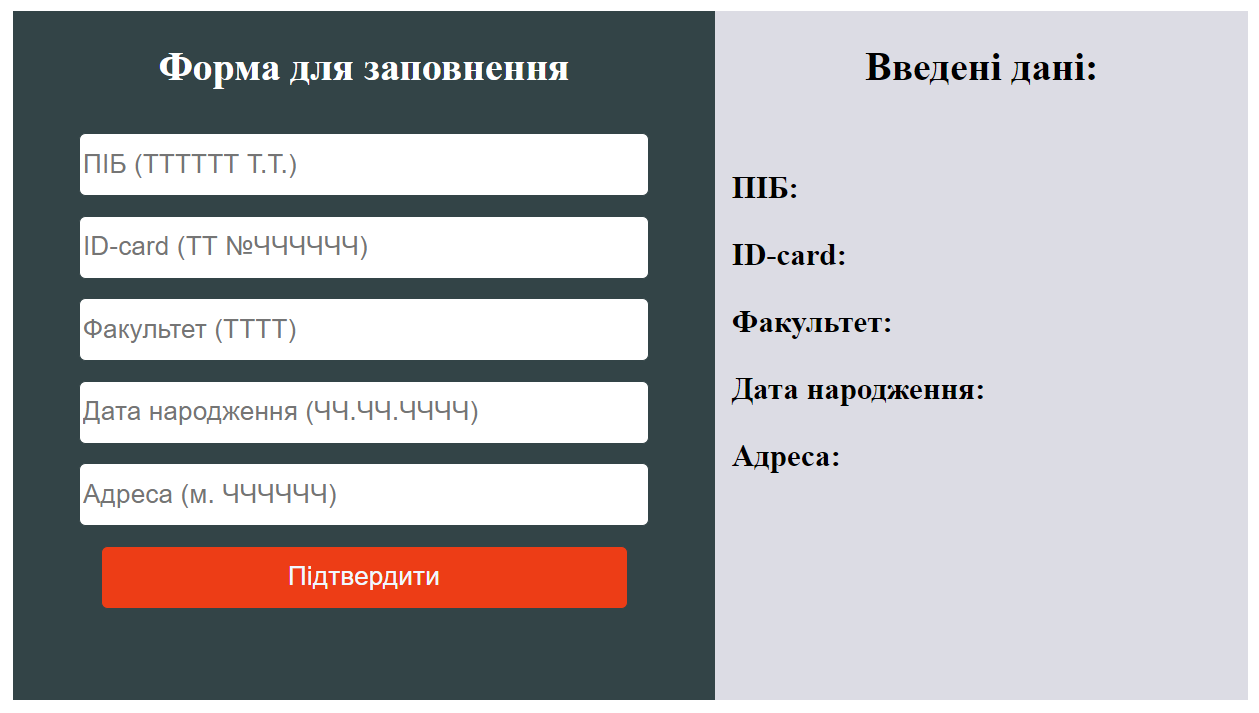


2) Також створимо базову структуру для полів виводу правильно заповнених даних.



3) Додамо стилі в окремому style.css файлі (Цього разу не було використано таких стилів які б не були використані в минулих роботах, тому в скрішоті немає сенсу)

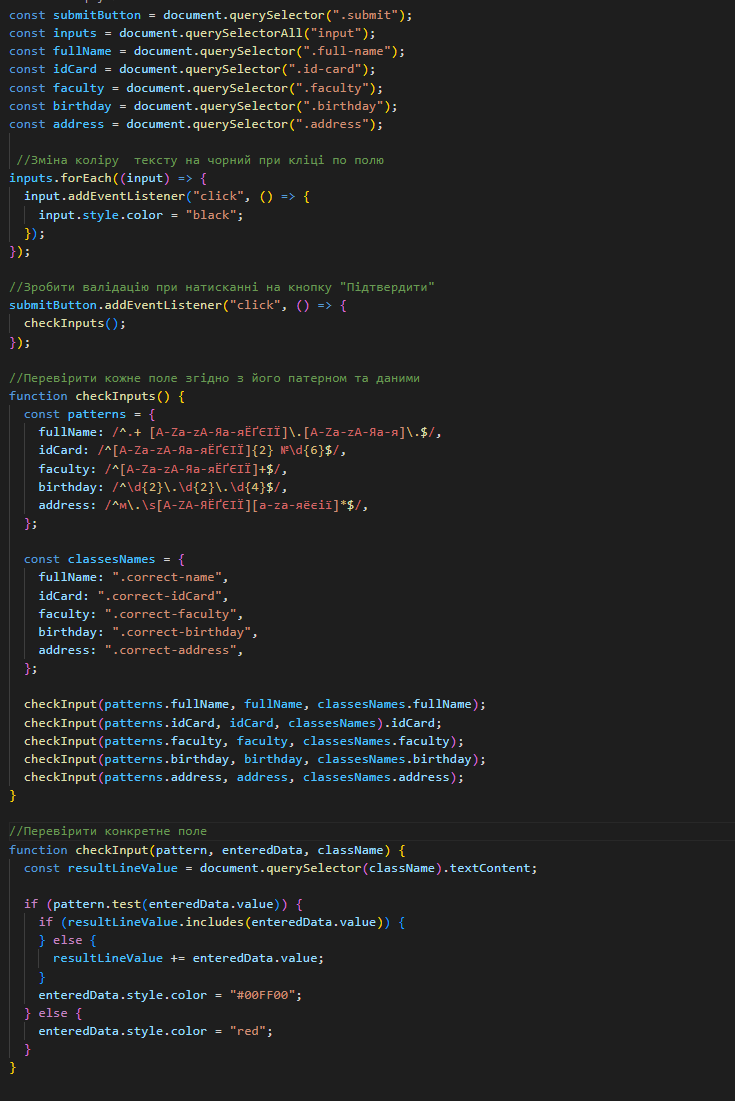
Проміжний результат:



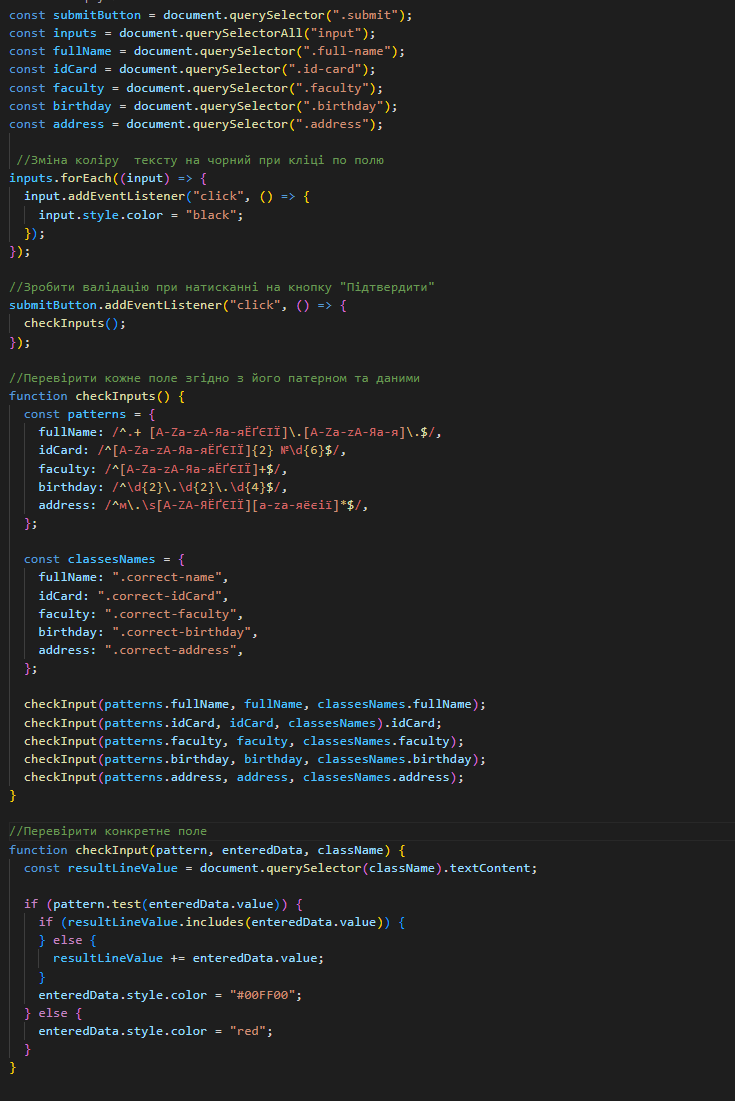
4) Напишемо скрипт, який буде задовольняти такі умови та виконувати такі функції:

* Перевірятиме валідність даних введених у поля.
* Враховувати що кожне поле має свій окремий «паттерн» заповнення
* Якщо введені дані співпадають із патерном, виводити їх у віконце праворуч при цьому підсвічуючи текст зеленим коліром
* Якщо введені дані не співпадають із патерном, підсвітити їх червоним кольором та не виводити їх у віконце
* Урахувати те, щоб при повторному введені правильних данних, вони не виводилися другий раз
* Змінювати колір тексту на чорний (замість червоного або зеленого) якщо користувач зробить клік по полю для вводу.

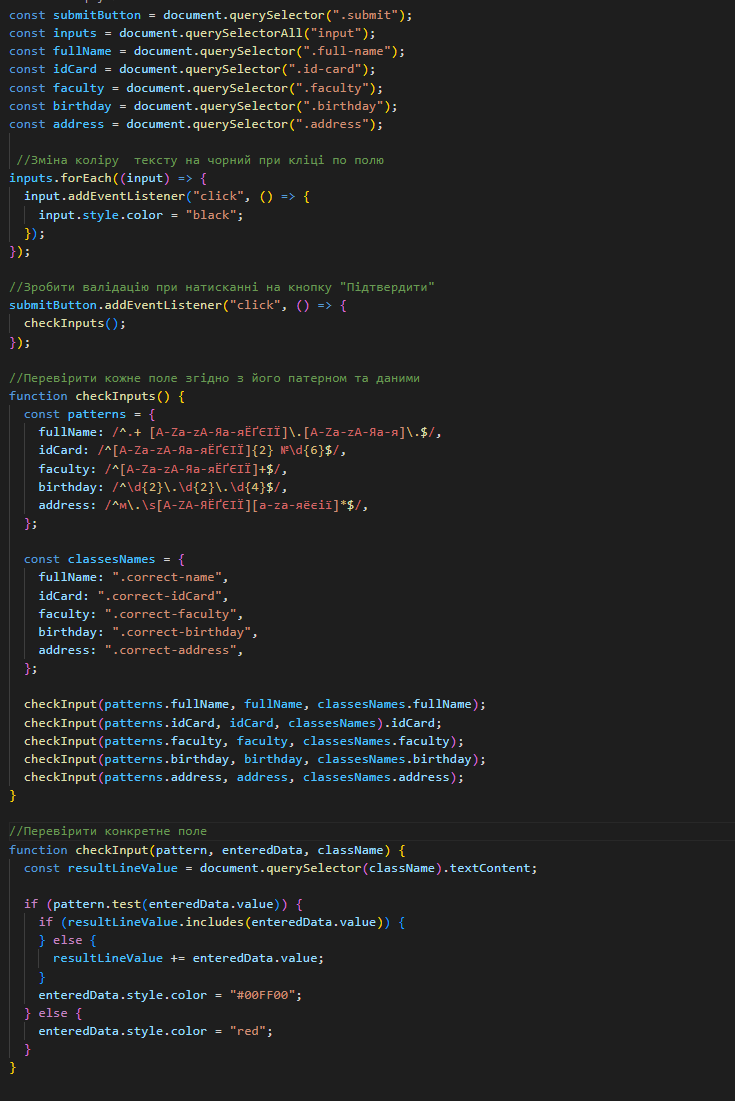
Напишемо функцію яка буде валідувати кожне окреме поле. Вона працює так: приймає в себе патерн, введені дані та ім’я класу (тобто саме те поле яке треба перевірити). Після цього робить перевірку, якщо валідація пройдена текст стану зеленем, якщо ні- червоним.



Після цього створимо функцію яка буде запускати перевірку для кожного з полів

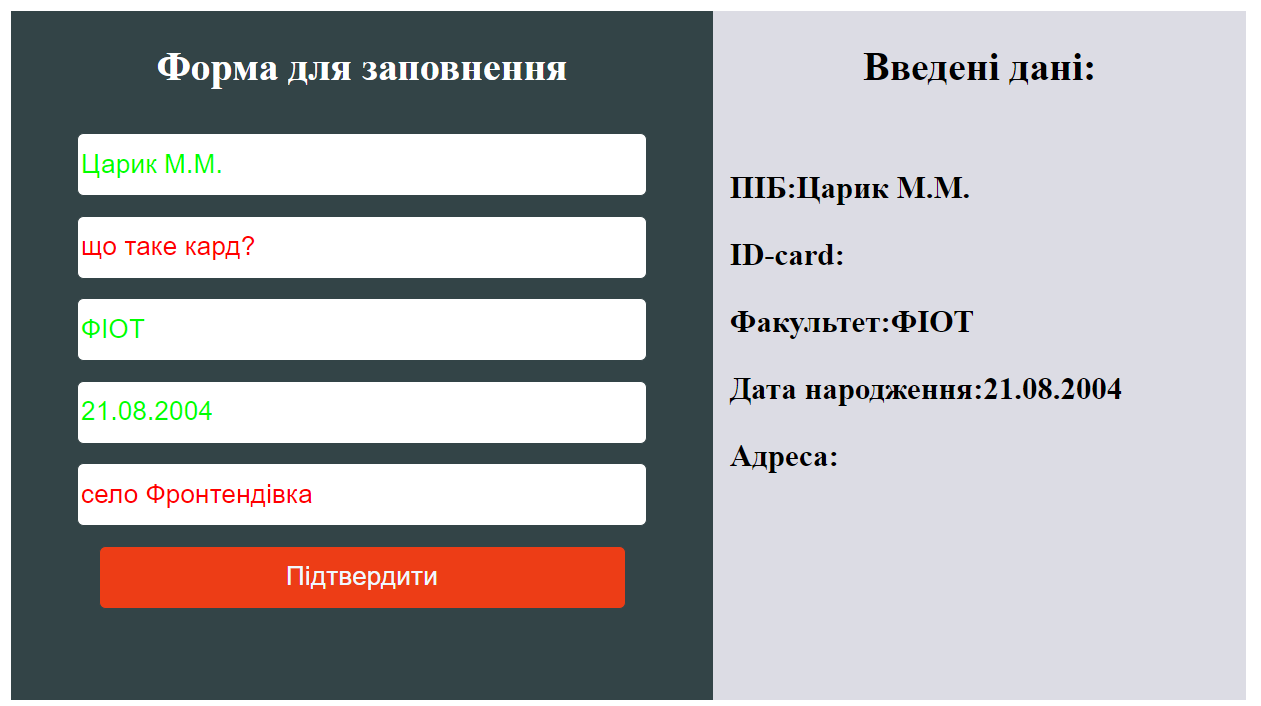


Крім того додамо обробку кнопки підвтердити та зміну кольору н ачорний при повторному введені. Тут використовуємо метод массиву .forEach() щоб пробіжатися по всім полям та змінити колір там де був клік

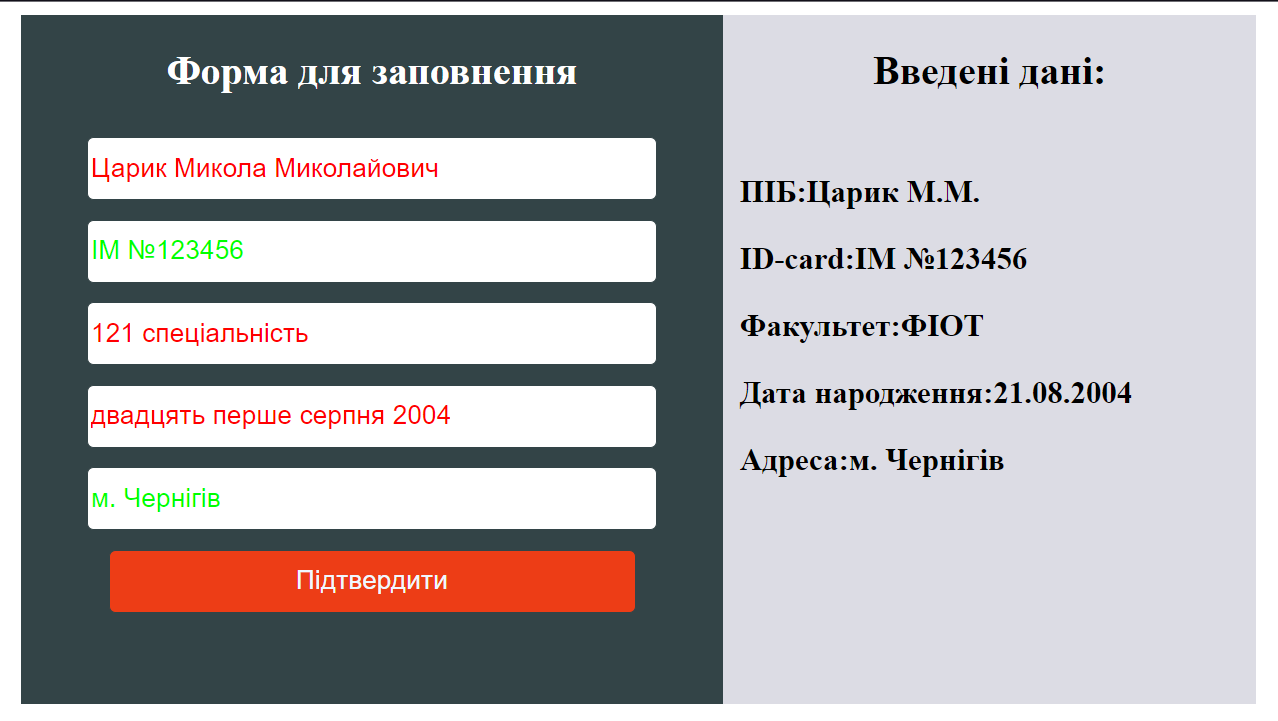


**Протестуємо програму:**

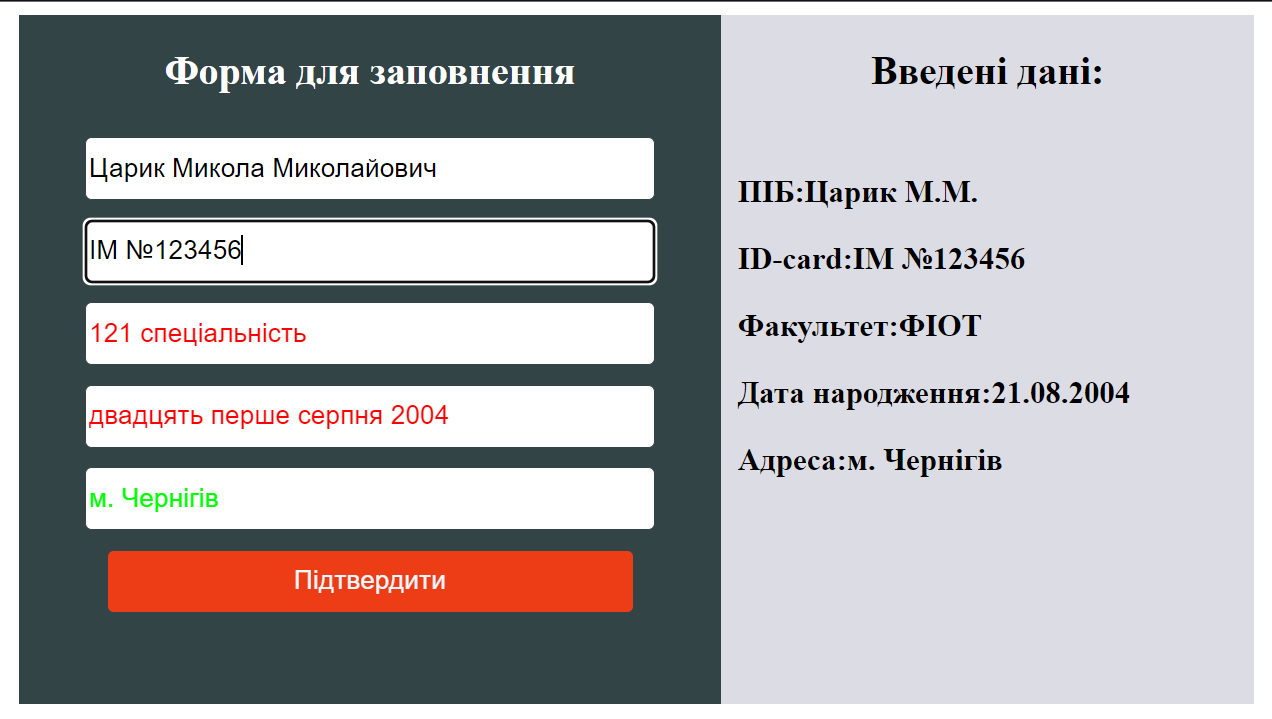
1) Заповнимо три поля вірно і два поля не вірно



2) Тепер навпаки, ті що були вірні зробимо невірними, а ті що невірні – вірними

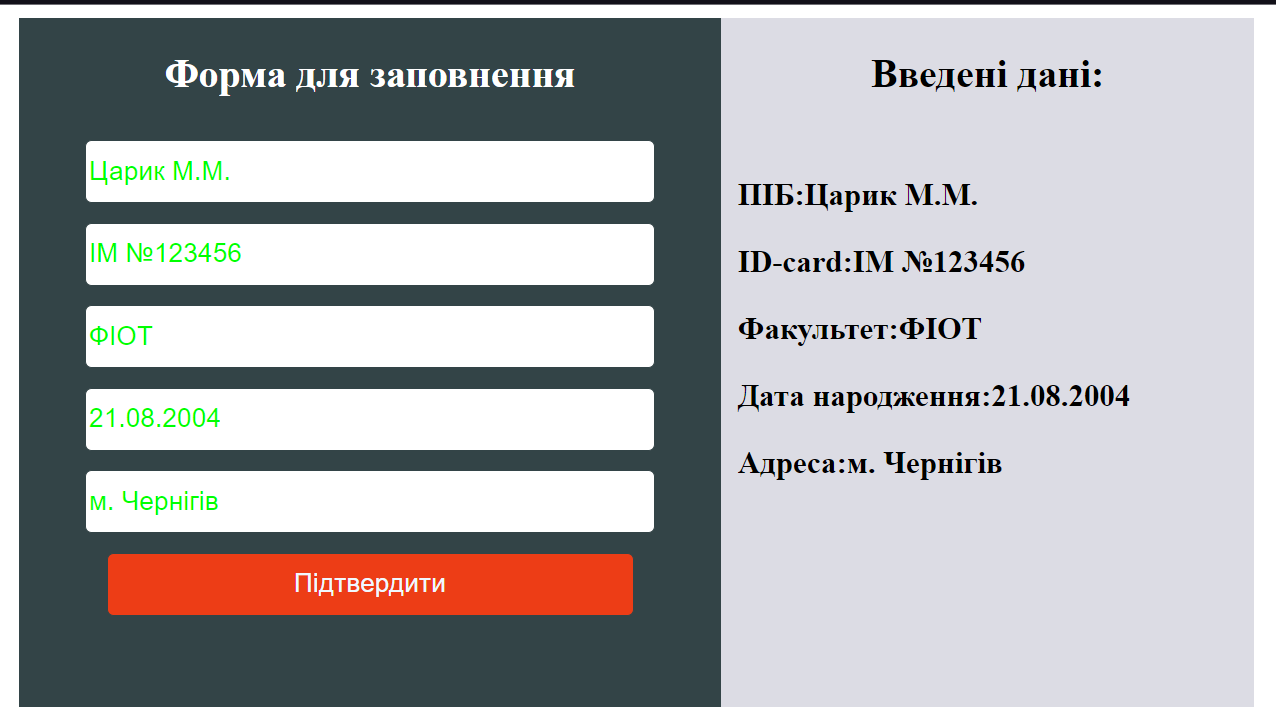


Як бачимо, тепер червоним горять вже інші невірні поля, а у віконці зберіглися правильні варінти (тобто ті, які я вводив в перший раз).



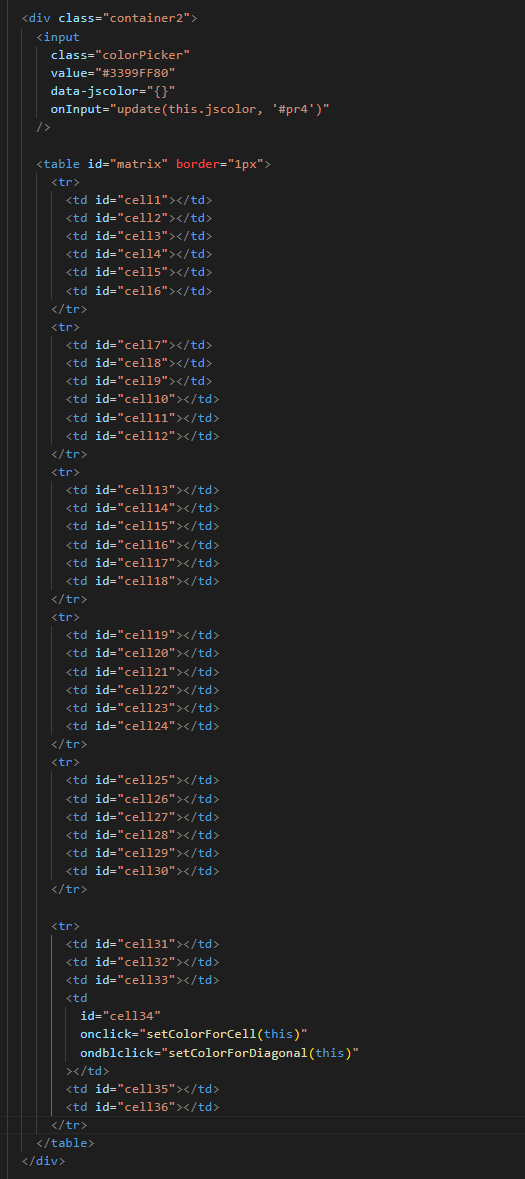
Бачимо що при натисканні на поле, текст чтає чорним натякаючи користувачеві що його можна редагувати.

3) Введемо всі правильні варіанти та впевнемося що у віконці праворуч відображатиметься тільки одна відповідь для кожного поля, як і повинно бути.

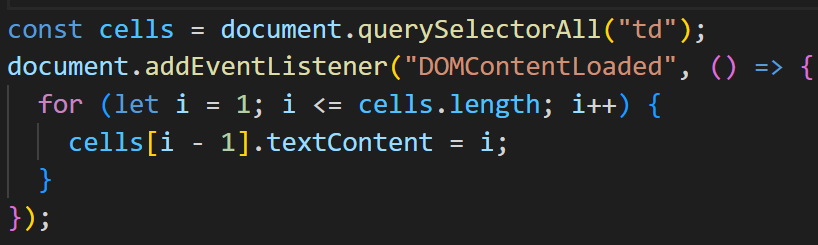


**Завдання 2**

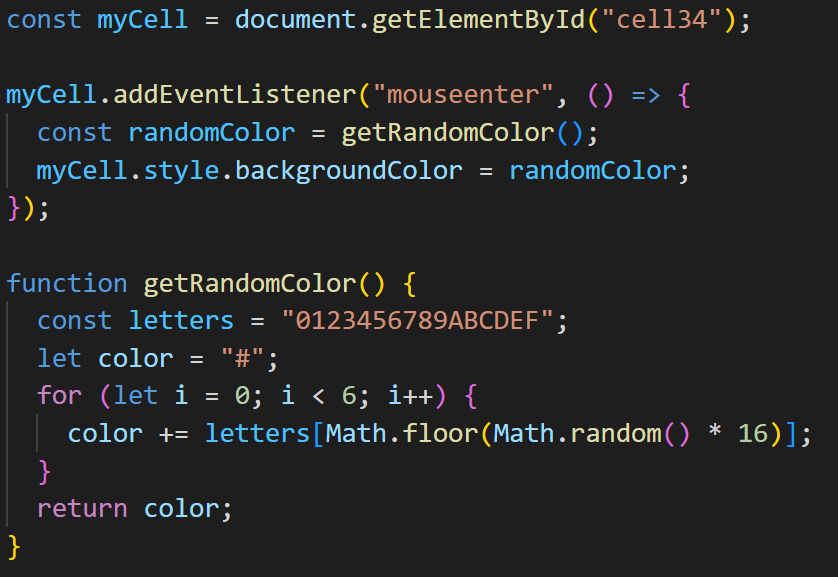
1) Створимо базову структуру матриці 6 на 6. Зробимо це таблицею за допомогою шести таких блоків



2) Напишемо скрипт який буде заповнювати таблицю числами від 1 до 36. Пишемо і-1 бо індекси масиву починаються з нуля, а ми хочему числа від 1 до 36



3) Додамо скрипт який буде зафарбовувати клітинку номер 34 у рандомний колір при наведенні. Рандомний колір створюємо за допомогою конкатинації рядків а також методів random(для випадкового числа) та floor (для округлення до меншого) з бібліотеки Math.

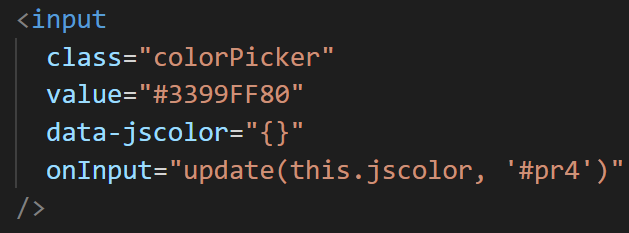


3) Тепер треба додати на сторінку палітру для вибору кольору (для цього я скористався бібліотекою jscolor.js) а також зробити так щоб після вибору кольору клітинка зафарбовувалася у цей колір при натисканні.

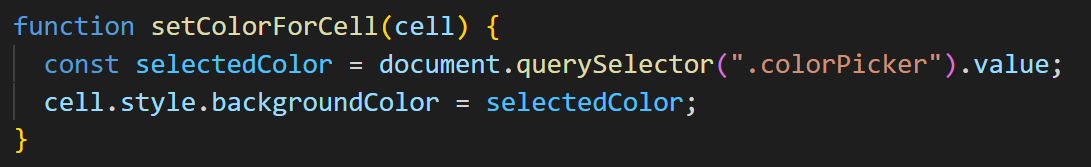
jscolorpicker.js - це бібліотека JavaScript, яка надає можливість створення інтерактивних елементів вибору кольору на веб-сторінках. Ця бібліотека дає змогу користувачам обирати кольори за допомогою графічного інтерфейсу, наприклад, шляхом переміщення курсору по палітрі або введення RGB-значень

Її використання:

Цей інпут являє собою палітру кольорів



А ось так ми можемо взяти колір з цієї палітри.

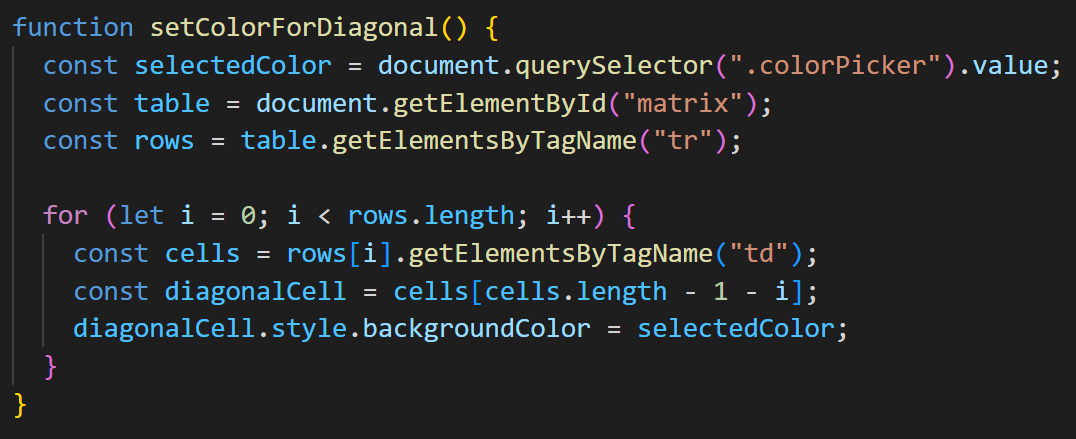


Було кілька альтернативних бібліотек які могли бути використані замість **jscolorpicker.js**.:

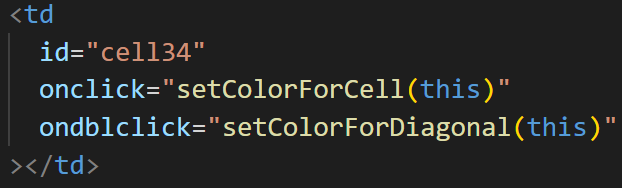
1. **Spectrum** - це популярна бібліотека для вибору кольорів, яка надає багато можливостей налаштувань і гарний користувацький інтерфейс.
2. **Pickr**- легковагий і красивий вибір кольору, який має підтримку різних форматів кольору та можливість налаштування.

Я обрав цей інструмент бо він легкий в освоєнні та потребує підключення лише одного додаткового файлу js. Для быльш детального ознайомлення можна відвідати їх офіційний сайт: https://jscolor.com

4) Додамо також функціонал зафарбовування побічної діагоналі матриці при подіїї подвійного натискання.



5) Прив’яжемо ці функії до нашої клітинки за допомогою onClick() та onDoubleClick()

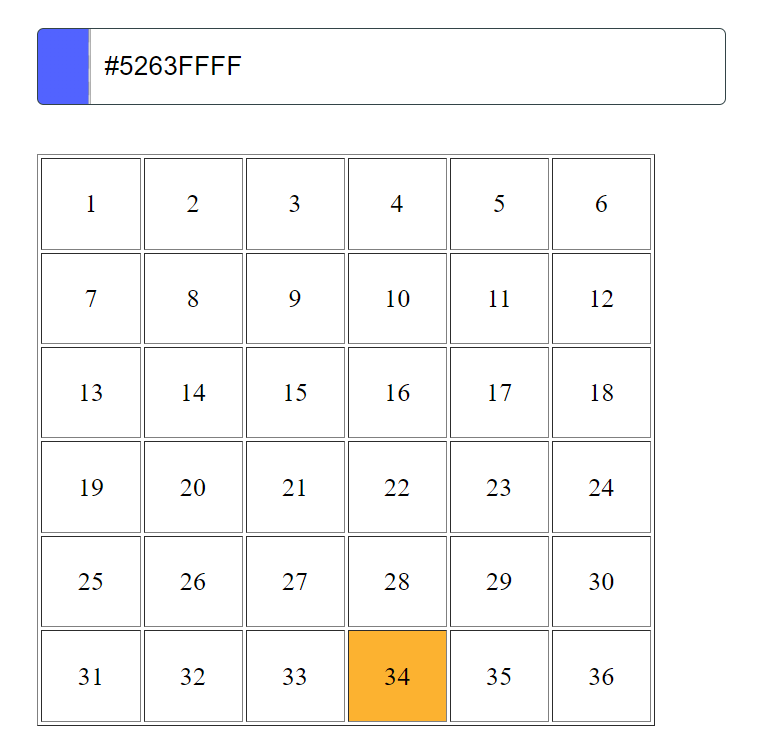


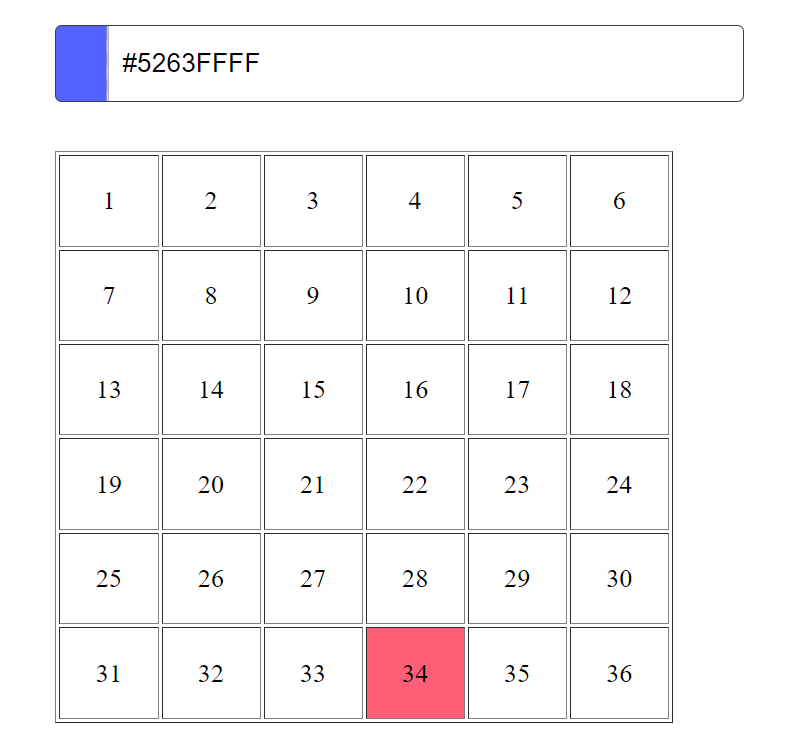
**Протестуємо програму:**

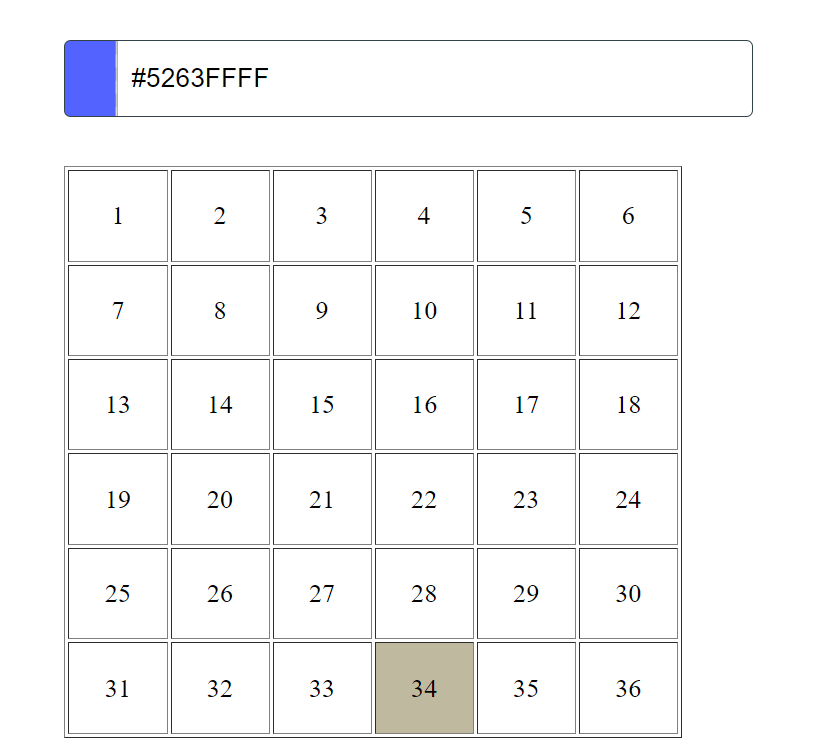
1) Початковий стан



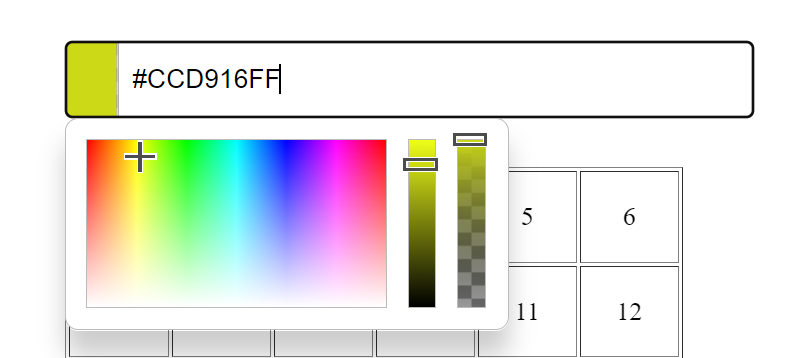
2) Тричі наведемо мишею по клітинці

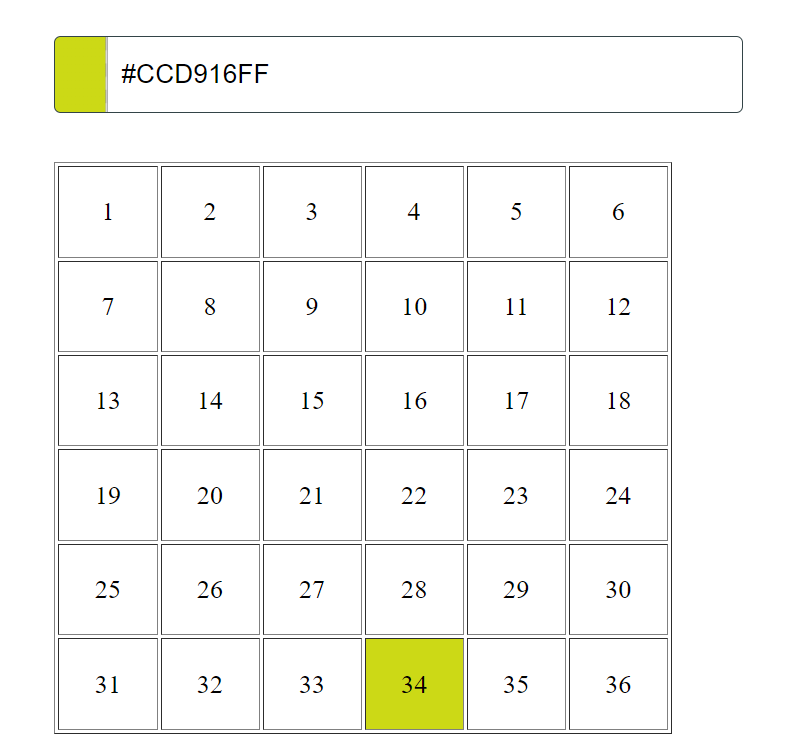




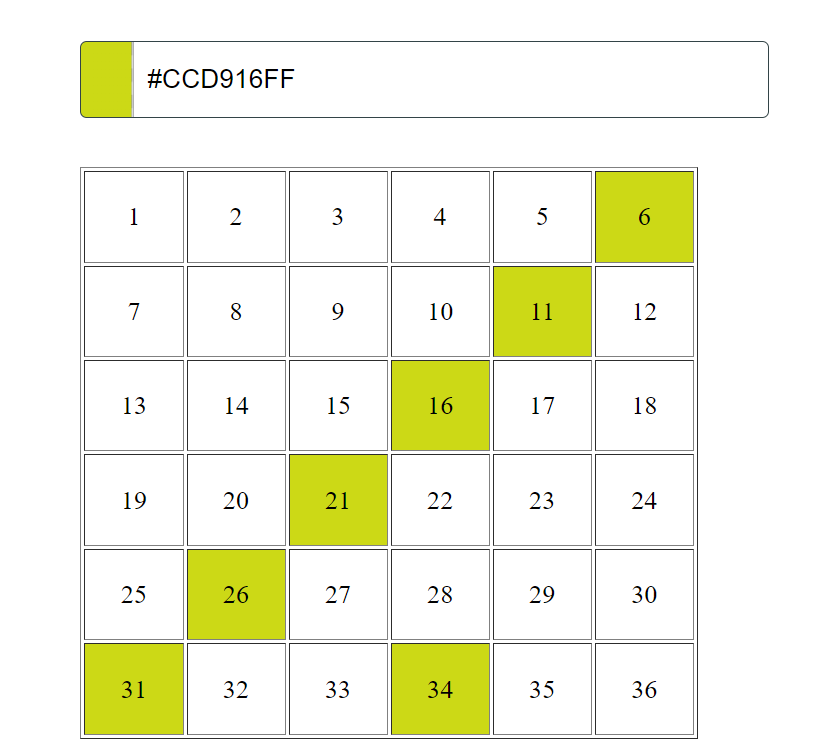


3) Оберемо колір та зробимо клік по нашій клітинці





5) Тепер зробимо дабл клік по клітинці, тим самим зафарбувавши побічну діагональ



**Висновок**

Під час лабораторної роботи були вирішені два завдання. Перше завдання передбачало взаємодію з таблицями та формами, де я вивчив роботу з подіями та використання регулярних виразів для валідації даних. Друге завдання стосувалося роботи з таблицею, де я змінював кольори клітинок за допомогою подій миші. Робота над цією лабораторною роботою допомогла мені освоїти обробку подій , роботу з допоміжними бібліотеками та взаємодію з елементами веб-сторінки за допомогою JavaScript.

**Додаток: (Посилання на репозиторій)**

<https://github.com/TsNikolay/Front-end-Basics-Labs>