

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут ІКНІ
Кафедра ПЗ**

ЗВІТ

До лабораторної роботи № 3

З дисципліни: *“Програмування в Інтернет”*

На тему: *“Особливості програмування мовою JavaScript та каскадні стилі.”*

Лектор:

проф. каф. ПЗ
Мельник Р. А.

Виконала:

ст. гр. ПЗ-26
Гуль О. М.

Прийняла:

асист. каф. ПЗ
Майхер В. Ю.

« ____ » _____ 2021 р.

Σ= ____

Тема роботи: Особливості програмування мовою JavaScript та каскадні стилі.

Мета роботи: Оволодіти структурою мови JavaScript для створення інтерактивних web-сайтів та різними способами розміщення коду JavaScript.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Одним з інструментів підтримки динамічних сценаріїв при перегляді Web-сторінок у межах комп'ютера користувача є мова програмування JavaScript. Нею забезпечується рух об'єктів на сторінці, введення та виведення параметрів, зміна зображень вікон тощо. Програми модифікації, створення гіпертекстових сторінок традиційно називають скриптами (scripts), які інтерпретуються програмою перегляду. Спосіб базується на ідеології об'єктно-орієнтованого програмування. Зупинимось на скриптах, написаних мовою JavaScript.

Мова програмування для реалізації інтерактивності web-сторінок: введення тексту, реакції на події, отримання інформації з сервера, здійснення обчислень, підлягає стандартизації, але не підтримується всіма браузерами. Javascript інтерпретується на браузері та інтегрується за змістом HTML/CSS.

Важливим елементом мови є події, які використовуються для виконання частин програмного коду скрипту, наприклад, `onLoad="Scroll();"`. До найвживаніших можна зарахувати такі:

- `onLoad` – виконання скрипту чи функції під час завантаження;
- `onChange` – породжується під час зміни значення елемента форми;
- `onClick` – породжується під час вибору об'єкта (button, checkbox і т.п.);
- `onSelect` – породжується під час вибору текстового об'єкта (text, textarea);
- `onSubmit` – під час натискання на кнопку Submit;
- `onUnload` – під час переходу до іншої сторінки.

Усі операції в програмі на JavaScript описують дії над об'єктами – елементами робочої області броузера та контейнерами мови HTML. Об'єкти мають властивості та методи. Існують також інші функції, які дають змогу працювати зі стандартними математичними функціями та керувати процесом виконання програми. JavaScript має механізм опрацювання подій перегляду динамічних об'єктів та управління багатовіконним інтерфейсом.

Засоби каскадного стилю призначені для опису, подання, планування та вигляду інформації на web-сторінці, на противагу мові HTML, призначеній для опису змісту. Стиль називається каскадним через керування атрибутами елементів у каскадному порядку:

- стилі браузера за замовчуванням;
- зовнішні файли стилів (`<link>`);
- внутрішні стилі (в межах тега `<style>` у заголовку);
- рядковий стиль (атрибути стилів елементів HTML).

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

1. На раніше розробленій сторінці із полем для гри у шахи:

- стилізувати сторінку різними способами формування стилів;
- додати до валідації форми даних гравця валідацію за допомогою JavaScript;
- додати кнопку старт, після натискання якої з'являтиметься таймер із активним відліком;
- за допомогою бібліотеки Chart.js, побудувати довільно обрану гістограму яка відображатиме наявну кількість кожного із типів фігур.

ХІД ВИКОНАННЯ

1. Стилїзувала сторінку різними засобами оформлення стилів

styles.css

```
.cell {
  width: 4em;
  height: 4em;
  position: relative;
}

.black {
  background-color: #ff645a;
  color: #d2d2d2;
}

.white {
  background-color: #d2d2d2;
  color: #ff645a;
}

.box{
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  margin-left: 0.15em;
}

.bottom-box{
  position: absolute;
  right: 0;
  bottom: 0;
```

```

    margin-right: 0.15em;
}

.top{
    z-index: 9;
}

img {
    max-width: 100%;
    max-height: 100%;
    margin: auto;
}

.center {
    text-align: center;
}

p {
    font-size: 1.5em;
}

span {
    font-weight: bold;
}

.flex-div{
    justify-content: space-around;
    margin-top: 5em;
    display: flex;
}

.vertical-flex {
    flex-direction: column;
}

canvas {
    max-width: 50%;
    max-height: 50%;
}

```

2. Додала до валідації форми даних гравця валідацію за допомогою JavaScript

Код завдання:

```

function validation(firstName, lastName, age){
    let message = ""
    if (firstName.length === 0){
        message += "First name can't be empty!\n"
    }

    if (! (/^[A-Za-z ]{1,50}/.test(firstName))){
        message += "First name must contain only uppercase and lowercase english letters  
and be no longer" +
            " than 50 symbols\n"
    }

    if (lastName.length === 0){
        message += "Last name can't be empty!\n"
    }
}

```

```

if (! (/^[A-Za-z ]{1,50}/.test(lastName))) {
    message += "Last name must contain only uppercase and lowercase english letters  
and be no longer" +
        " than 50 symbols\n"
}

if (age === '0') {
    message += "The age isn't chosen!"
}

window.alert(message)
}

```

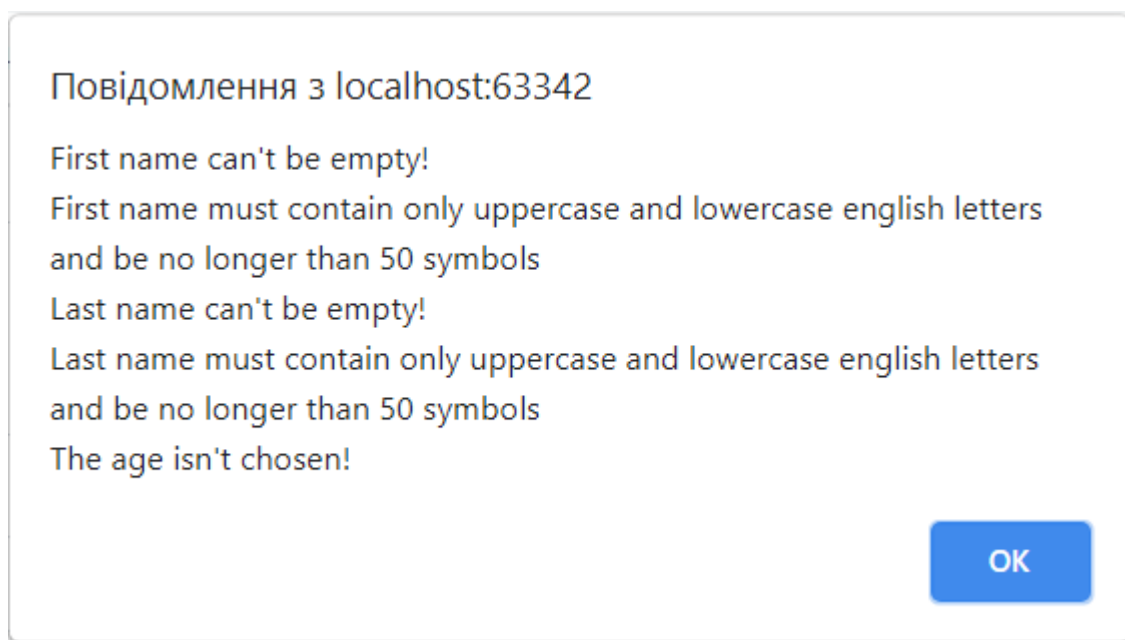
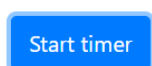


Рис. 1. Валідація за допомогою JavaScript

3. Додала кнопку старт, після натискання якої з'являтиметься таймер із активним відліком



00 : 08

Рис. 2. Кнопка старт і таймер

4. За допомогою бібліотеки Chart.js, побудувала гістограму яка відображатиме наявну кількість кожного із типів фігур.

Код завдання:

```
function createChart(){
  let ctx = document.getElementById('myChart').getContext('2d');
  ctx.width = 800
  ctx.height = 600
  let chessNames = ['Pawn', 'Bishop', 'Knight', 'Rook', 'Queen', 'King']
  let chess = getChessCounts()
  let chessCounts = []

  for(let i = 0; i < chess.length; i++){
    chessCounts.push(chess[i]["count"])
  }

  let chart = new Chart(ctx, {
    // The type of chart we want to create
    type: 'bar',

    // The data for our dataset
    data: {
      labels: chessNames,
      datasets: [{
        label: 'Chess figures and their counts on the field',
        backgroundColor: [
          'rgba(255, 99, 132, 0.2)',
          'rgba(54, 162, 235, 0.2)',
          'rgba(255, 206, 86, 0.2)',
          'rgba(75, 192, 192, 0.2)',
          'rgba(153, 102, 255, 0.2)',
          'rgba(255, 159, 64, 0.2)'
        ],
        borderColor: [
          'rgba(255, 99, 132, 1)',
          'rgba(54, 162, 235, 1)',
          'rgba(255, 206, 86, 1)',
          'rgba(75, 192, 192, 1)',
          'rgba(153, 102, 255, 1)',
          'rgba(255, 159, 64, 1)'
        ],
        borderWidth: 1,
        data: chessCounts
      }]
    },

    options: {}
  });
}

function getChessCounts(){
  let chessLetters = ['p', 'b', 'n', 'r', 'q', 'k']
  let chess = []
  for (let i = 0; i < chessLetters.length; i++){
    chess.push({
      letter: chessLetters[i],
      count: 0
    })
  }
}
```

```

    })
  }

  let images = document.getElementsByTagName('img')
  for (let i = 0; i < images.length; i++){
    let src = images[i].getAttribute('src')
    let letter = src[8]
    chess.find(function (element){
      return element['letter'] === letter
    }).count += 1
  }

  return chess
}

```

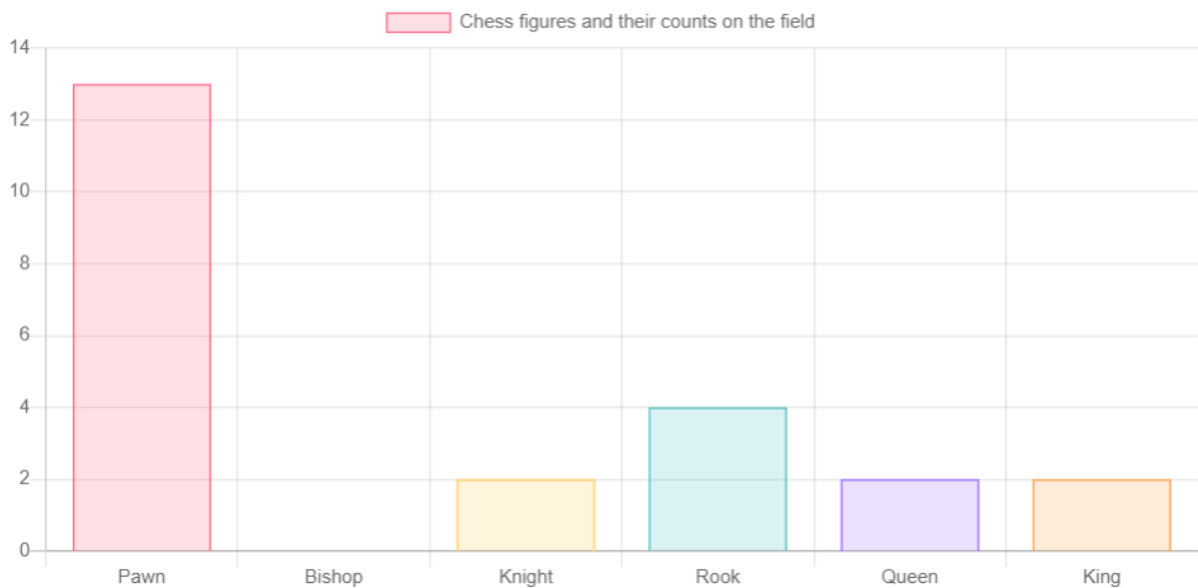


Рис. 3. Гістограма

ВИСНОВКИ

На лабораторній роботі я оволоділа структурою мови JavaScript для створення інтерактивних web-сайтів та різними способами розміщення коду JavaScript. Попрацювала з каскадними стилями, стилізувавши попередньо розроблену веб сторінку. За допомогою мови JavaScript написала валідацію для полів введення даних гравця і таймер з активним відліком. Ознайомила з бібліотекою Chart.js, побудувала гістограму з кількостями шахових фігур на дошці.