

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІКТА  
кафедра ЗІ



**ЗВІТ**

**До лабораторної роботи № 5**

з курсу: «Веб-програмування»  
на тему: «Розгортання статичного сайту»

**Виконав:**  
ст. гр. КБ-301  
Дмитрук В. Д.

**Прийняв:**  
Петрів П. П.

Львів – 2025

**Мета:** ознайомитися з основними поняттями та синтаксисом JavaScript. Навчитися оголошувати змінні, працювати з базовими типами даних та операціями над ними. Засвоїти основні структури керування програмою (умовні оператори, цикли). Навчитися взаємодіяти з елементами HTML (DOM) за допомогою JavaScript. Зрозуміти основи обробки подій (Events) у браузері.

## Теоретичні відомості (відповіді на питання)

### 1. Способи оголошення змінних у JavaScript та їх відмінності

- `var` — старий спосіб, має **функціональну область видимості** (function scope) і дозволяє **повторне оголошення**.
- `let` — блочна область видимості (block scope), **не дозволяє повторно оголошувати** змінну в межах одного блоку.
- `const` — як `let`, але значення **не можна змінити** (константа).

### 2. Типи даних у JavaScript

#### Прості (Primitive):

- `number`, `string`, `boolean`, `null`, `undefined`, `bigint`, `symbol`

#### Складні (Reference):

- `object`, `array`, `function`

*Відмінність:* Примітиви зберігаються і копіюються **за значенням**, об'єкти — **за посиланням**.

### 3. Різниця між `==` і `===`

- `==` — **нестроге порівняння**, виконує **неявне приведення типів**.
- `===` — **строге порівняння**, **без приведення типів**.

#### Приклад:

```
'5' == 5    // true
'5' === 5   // false
```

### 4. Основні структури керування у JavaScript

- Умови: `if`, `else`, `switch`
- Цикли: `for`, `while`, `do...while`, `for...of`, `for...in`
- Інші: `try...catch`, `return`, `break`, `continue`

### 5. Види функцій

#### Function Declaration:

```
function greet() {
  console.log("Hello");
}
```

### Function Expression:

```
const greet = function() {  
  console.log("Hello");  
};
```

### Arrow Function:

```
const greet = () => {  
  console.log("Hello");  
};
```

### Відмінності:

- **Declaration** — піднімається (hoisted).
- **Expression** і **Arrow** — не піднімаються.
- **Arrow** не має власного **this**.

## 6. Методи обробки масивів

- **forEach** — виконує функцію для кожного елемента.
- **map** — повертає новий масив, змінюючи кожен елемент.
- **filter** — повертає новий масив з елементами, що відповідають умові.
- **reduce** — агрегує масив до одного значення.
- **find** — повертає перший елемент, що відповідає умові.
- **some / every** — перевіряють умови для частини або всього масиву.

## 7. Об'єкти в JavaScript

### Створення:

```
let obj = {  
  name: "Ivan",  
  age: 25  
};
```

### Зміна:

```
obj.age = 26;
```

### Доступ:

```
console.log(obj.name); // через крапку  
console.log(obj["age"]); // через дужки
```

## 8. Робота з DOM

### Отримання елементів:

```
document.getElementById("id");  
document.querySelector(".class");
```

### Зміна властивостей:

```
element.textContent = "Новий текст";  
element.style.color = "red";  
element.setAttribute("title", "підказка");
```

## 9. Події та слухачі

**Події** — дії користувача (клік, клавіша, наведення тощо).

### Додавання слухача:

```
element.addEventListener("click", function() {  
    alert("Клік!");  
});
```

### Основні події:

- `click`, `mouseover`, `keydown`, `submit`, `load`, `change`

## 10. Навіщо потрібен `console.log` і які є інші інструменти?

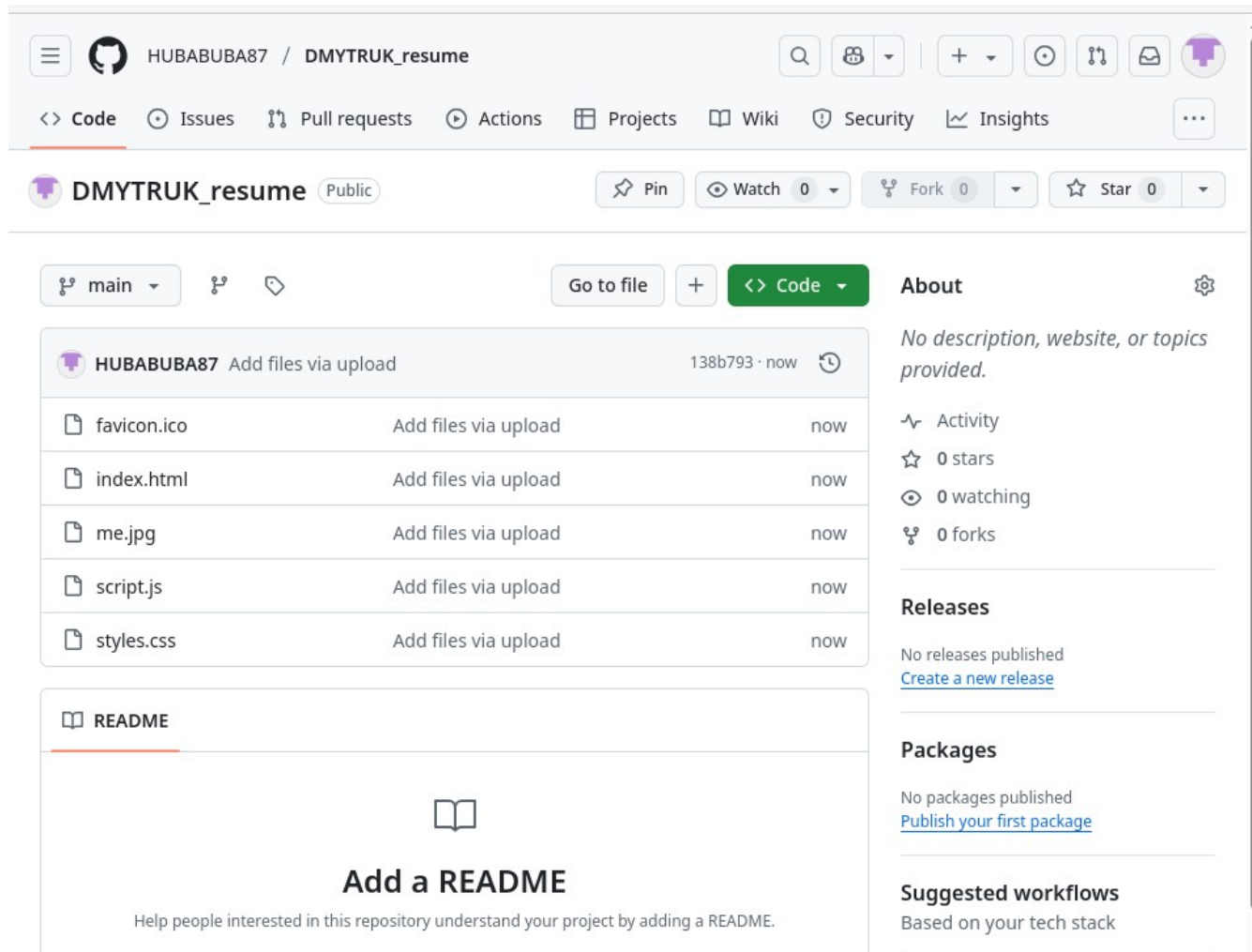
- `console.log` — виводить значення в консоль, допомагає відстежувати стан коду.
- Інші:
  - `console.error`, `console.warn`, `console.table`
  - Інструменти розробника браузера (DevTools)
  - Точки зупину (breakpoints)
  - Вкладка Network для відстеження запитів

## Хід роботи

1. Створити акаунт на сайті `github.com`
2. Створити порожній публічний репозиторій та завантажити в нього свій веб сайт(папка із кодом). Посилання на репозиторій та його скріншот додати до лабораторної роботи.
3. Створити акаунт на сайті `vercel.com` та згідно з інструкціями опублікувати власний сайт в мережі. Посилання на опублікований сайт та його скріншот додати до звіту лабораторної роботи.

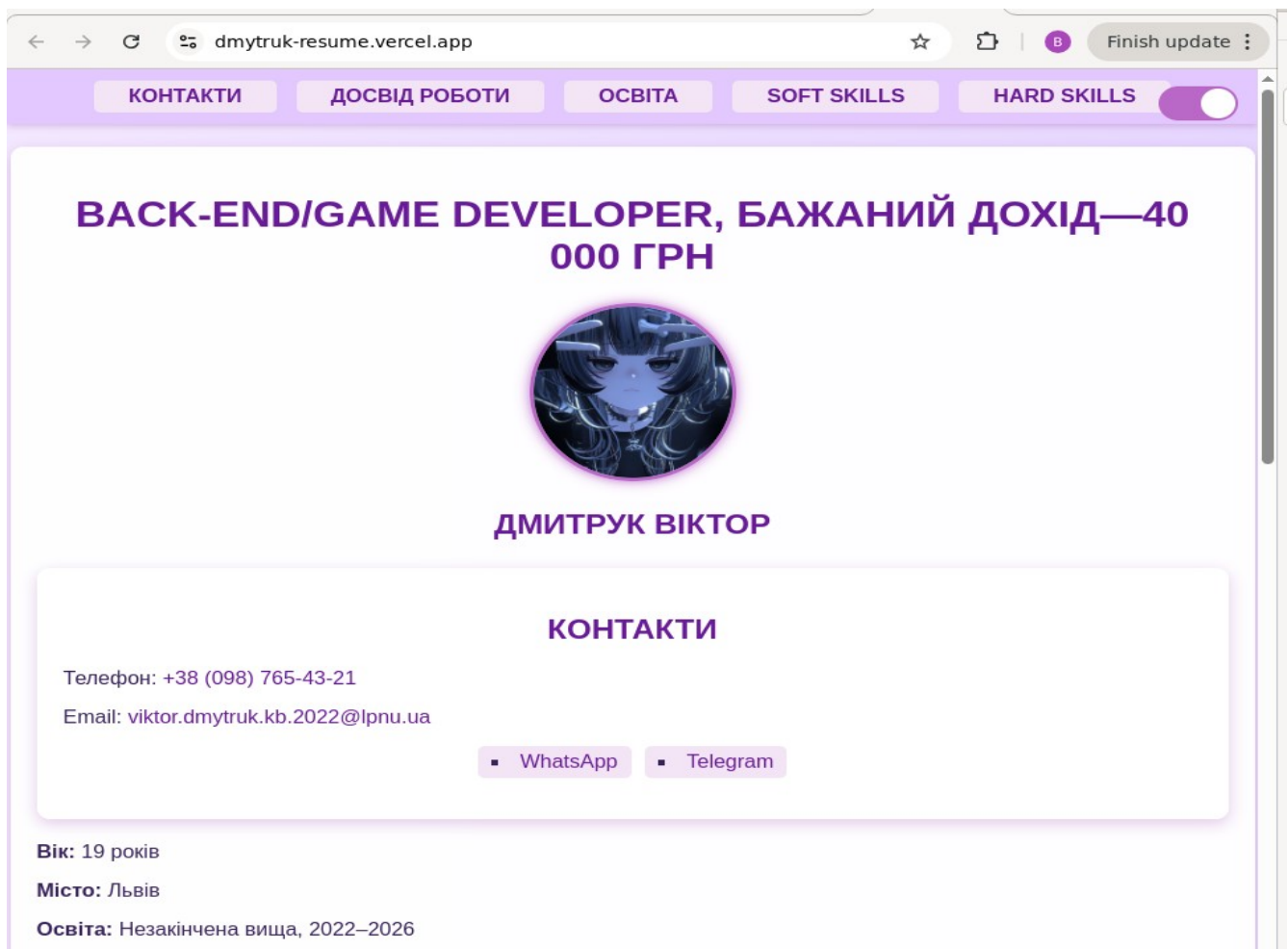
4. Зробити будь-які зміни вихідного коду, що змінюють зовнішній вигляд сайту та закомітати їх, посилання на ваш коміт додати до лабораторної роботи. Переконайтеся, що після коміту опублікований сайт автоматично оновився.
5. (Опційно) Придбати власний домен та спробувати розібратися як підключити його до свого сайту.

## Скріншот репозиторію



Посилання на репозиторій: [https://github.com/HUBABUBA87/DMYTRUK\\_resume](https://github.com/HUBABUBA87/DMYTRUK_resume)

Посилання на сайт: <https://dmytruk-resume.vercel.app/>



Міняємо зарплату на гітхабі в репозиторії в файлі index.html.

```
<header>
  <h1>Back-end/Game Developer, бажаний дохід—35 000 грн</h1>
  
  <h2><p><strong>Дмитрук Віктор</strong></p></h2>
```

Коміт:

[https://github.com/HUBABUBA87/DMYTRUK\\_resume/commit/15e8e1f09106c17a9af41628f96c272d8571791a](https://github.com/HUBABUBA87/DMYTRUK_resume/commit/15e8e1f09106c17a9af41628f96c272d8571791a)

1 file changed +1 -1 lines changed Search within code

index.html +1 -1

↑	@@ -24,7 +24,7 @@	
24	24	</nav>
25	25	
26	26	<header>
27	-	<h1>Back-end/Game Developer, бажаний дохід-40 000 грн</h1>
27	+	<h1>Back-end/Game Developer, бажаний дохід-35 000 грн</h1>
28	28	
29	29	<h2><p><strong>Дмитрук Віктор</strong></p></h2>
30	30	

Оновлений сайт (оновлено зарплату):

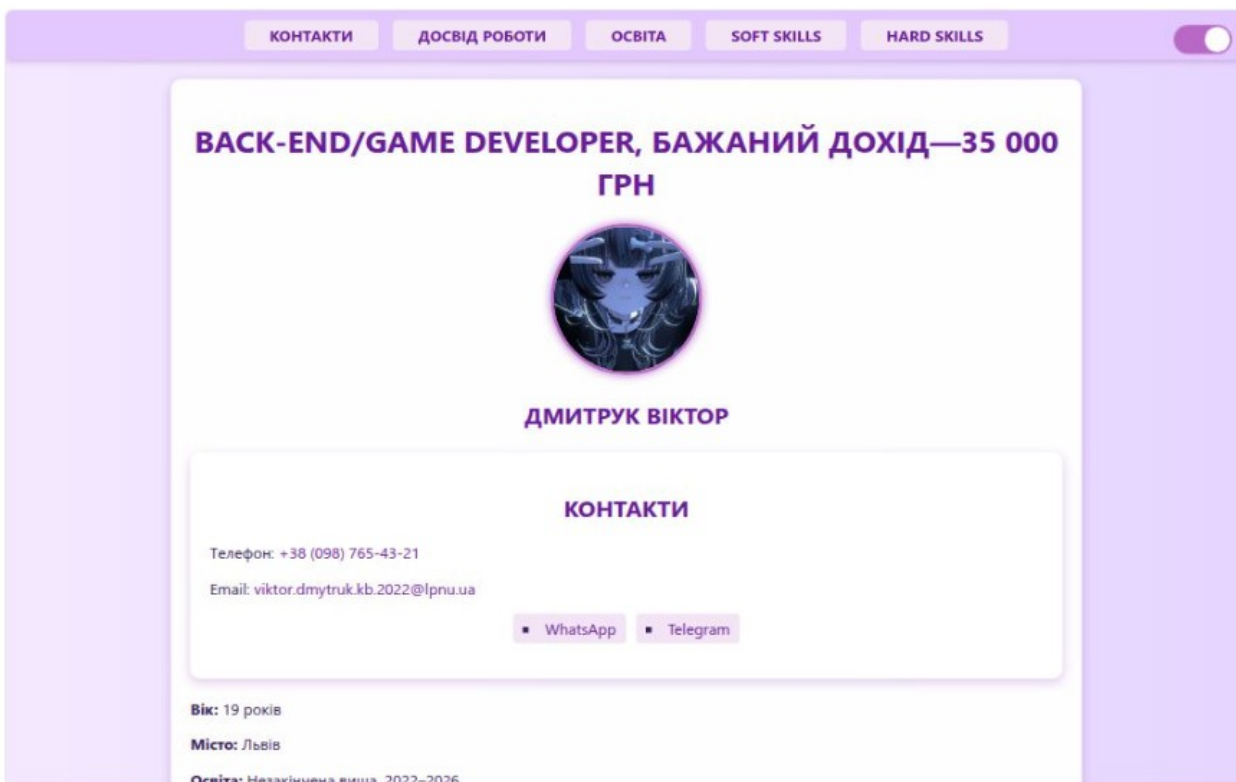
BACK-END/GAME DEVELOPER, БАЖАНИЙ ДОХІД—35  
000 ГРН



ДМИТРУК ВІКТОР

## Production Deployment

...



### Deployment

dmytruk-resume-5nbtnwriq-vals-projects-e1b49681.vercel.app



### Domains

dmytruk-resume.vercel.app [🔗](#) +2

### Status

### Created

● Ready

Just now by HUBABUBA87

**Висновок:** виконавши лабораторну роботу, я ознайомився з основними поняттями та синтаксисом JavaScript. Навчився оголошувати змінні, працювати з базовими типами даних та операціями над ними. Засвоїв основні структури керування програмою (умовні оператори, цикли). Навчився взаємодіяти з елементами HTML (DOM) за допомогою JavaScript. Зрозумів основи обробки подій (Events) у браузері.