 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №8**

з дисципліни «Веб-технології та веб-дизайн»

**Виконав:**

студент гр. БС-82

Нікіта Воронін

**Перевірив:**

ас. каф. БМК Матвійчук О. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

Створити 1 сторінковий сайт який має містити наступну інформацію:

Час з моменту останнього оновлення даних. (Бажано в header, та в секундах з точністю до 0.1)

Дані по позиціях на біржі для 8 найменувань (на ваш вибір)

Посилання на тест: [testmoz.com/2985692](https://testmoz.com/q/2985692)

Пароль до тесту: fbmi2020

**Пункт 1:**

Доступ до наних біржі відбувається за допомогою API описаного тут: <https://iexcloud.io/docs/api/>

Ключ доступу: pk\_ba9bb61a36a24ec3a5ae70362941f41c

Частота оновлень: 20 сек

p.s: даний ключ має задавитись конфігураційно та не повинен фігурувати на gitHub

**Пункт 2:**

Таблиця з результатами повинна містити наступні поля:

Назву компанії (Приклад: "Apple Inc.")

Назву позиції (Приклад: "AAPL")

Останню ціну позиції (Приклад: "158.73")

Різницю з моменту останнього оновлення (Приклад: "-1.67")

**При виконанні роботи обов'язково використовувати бібліотеку RXjs**

PS1: Потрібно чітко відділяти потоки роботи з даними

PS2: Статичні поля  необхідно запитувати лише 1 раз

Різницю в часі з моменту останньої угоди (Приклад: "2.9с") // постійно оновлювати

Виконання, Лістинг:  
**index.html**  
<!DOCTYPE html>

<html>

<head>

<meta Charset="utf-8">

<link rel ="stylesheet" href="resources/style.css">

<script src="https://unpkg.com/rxjs/bundles/rxjs.umd.min.js"></script>

</head>

<body>

<table id = "Table">

<caption>

<h3>

Час з останнього оновлення даних:

<p id="latestUpdate"></p>

</h3>

</caption>

<thead>

<tr>

<th>Назва позиції</th>

<th>Назва компанії</th>

<th>Остання ціна позиції</th>

<th>Різниця з моменту останнього оновлення</th>

</tr>

</thead>

<tbody id = "TableData">

</tbody>

</table>

</body>

<script src="resources/js.js"></script>

</html>

**Js.js**

let request = new XMLHttpRequest(); // Об'єкт для створення запитів

const key = prompt("Введіть ключ");

const urlConstRequest = 'https://cloud.iexapis.com/v1/stock/market/batch?symbols=MSFT,INTC,ATVI,EA,GOOGL,AMD,NFLX,TSLA&types=quote&filter=symbol,companyName,latestPrice,change&token=' + key;

const urlDinamicRequest = 'https://cloud.iexapis.com/v1/stock/market/batch?symbols=MSFT,INTC,ATVI,EA,GOOGL,AMD,NFLX,TSLA&types=quote&filter=latestPrice,change&token=' + key;

let latestUpdate\_Table = 0;

request.open('GET', urlConstRequest, true); // Перший запит на отримання всіх необхідних данних

request.onreadystatechange = function() // Отримання відповіді

{

if (request.readyState === XMLHttpRequest.DONE && request.status === 200){ //ReadyState DONE - операція повністю виконана , status 200 - без помилок

creat\_table(JSON.parse(request.response));

latestUpdate\_Table = Date.now();

}

}

request.send('');

rxjs.interval(20000) //Виклик оновлення данних

.subscribe(() => {

request.open('GET', urlDinamicRequest, true); // запити для оновлення всіх необхідних данних

request.onreadystatechange = function() // Отримання відповіді

{

if (request.readyState === 4 && request.status === 200) {

update\_table(JSON.parse(request.response),Date.now());

latestUpdate\_Table = Date.now();

}

}

request.send('');

});

rxjs.interval(100) //Таймер часу останнього оновлення

.subscribe(() => { document.getElementById('latestUpdate').textContent = ((Date.now() - latestUpdate\_Table) / 1000).toFixed(1); });

function creat\_table(data) //Загальне створення таблиці

{

let table = document.getElementById("TableData");

for (i in data)

{

let tr = document.createElement('tr');

let z = 0;

for (j in data[i].quote)

{

let td = document.createElement('td');

td.textContent = data[i].quote[j] + ((z>1)?" $":"");

tr.appendChild(td);

z++;

}

table.appendChild(tr);

}

document.getElementById('Table').appendChild(table);

}

function update\_table(data)

{

const rows = document.getElementById('TableData').rows;

let x = 0;

for (i in data)

{

let z = 2;

for (j in data[i].quote)

{

rows[x].cells[z].textContent = data[i].quote[j] + " $";

z++;

}

x++;

}

}

**Style.css**

html

{

font-family: calibri;

}

p

{

margin: 0;

font-size: 150%;

}

thead

{

background: #ccc;

}

table

{

border-collapse: collapse;

text-align: center;

margin:auto;

margin-top: 10vh;

border: 1px solid black;

}

th,td,tr

{

padding: 5px;

border-style: solid;

border-width: 0 1px 1px 0;

border-color: black;

}