

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 5
з дисципліни “Основи веб-програмування”

Виконав
студент II курсу
групи КП-93

Катрич Владислав Сергійович
(прізвище, ім'я, по батькові)

варіант №25

Перевірів
“_____” “_____” 20__ р.
викладач

Гадиняк Руслан Анатолійович
(прізвище, ім'я, по батькові)

Київ 2021

Мета роботи

Навчитись створювати динамічні веб-сторінки за допомогою браузерної мови JavaScript.

Навчитись працювати із технологією AJAX для отримання даних від REST API веб-сервера.

Використати WebSocket API для оповіщення сервером подій зміни даних.

Постановка завдання

Створити веб сайт із використанням динамічних можливостей браузерного JavaScript, AJAX та WebSocket API.

Веб сторінки можуть бути реалізовані окремо, або у вигляді однієї динамічної веб сторінки.

Реалізувати:

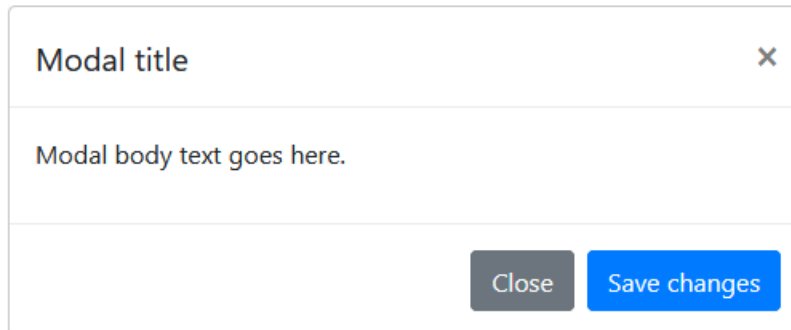
1. Сторінку створення сутності

1. Сторінка містить форму створення сутності
2. Замість стандартної поведінки форми відправляти AJAX запит до REST API сервера.
3. Отримати у відповідь всі дані створеної сутності із новим ідентифікатором.
4. Створити під формою простий список "Last created", у який додавати елемент сутності після отримання відповіді про її створення.
5. Кожен елемент списку має містити короткий опис (наприклад, назву сутності), дату додавання сутності і посилання на сторінку сутності.
6. Сортувати елементи у списку останніх доданих сутностей від найновішого до найстарішого.

2. Сторінку перегляду і видалення сутності

1. Сторінка містить дані сутності і кнопку видалення цієї сутності
-

2. Перед відправленням AJAX запиту до REST API на видалення показати користувачу нові графічні елементи з запитанням чи справді видалити вказану сутність і кнопки "Yes" і "Cancel". Для цього можна, наприклад, використати [модальні вікна](#).



3. Після отримання відповіді графічно показати, що сутність видалена.
3. **Сторінку перегляду і пошуку сутностей з пагінацією:**
 1. Веб сторінка містить пагінований список сутностей із номером сторінки, загальною кількістю сторінок, [кнопками для переходу по сторінках сутностей](#) (активні в залежності від можливості переходу по них), поле для вводу рядка пошуку по якому будуть знайдені сутності, кнопку пошуку і кнопку переходу до сторінки створення сутності. Кожен елемент списку сутностей містить короткий опис (назву) сутності і посилання на сторінку сутності.



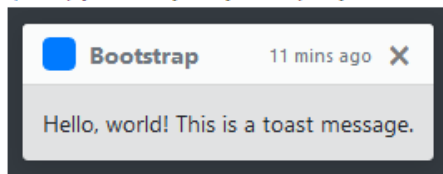
2. **Пагінація:** використати AJAX запити із динамічним оновленням вмісту поточної веб сторінки та показу сторінки даних. Дані першої сторінки сутностей завантажувати також динамічно після того як веб сторінка була отримана браузером і побудоване DOM дерево.
3. **Пошук:** використати AJAX запит до REST API сервера із динамічним оновленням вмісту поточної веб сторінки та показу результатів пошуку. Якщо пошук неуспішний, виводити відповідне повідомлення.
4. **Показ процесу виконання запиту і обробка помилок:**
 1. Виконання кожного із AJAX запитів повинно в першу чергу блокувати елементи, які спричинили цей запит (наприклад, зробити кнопку неактивною чи заховати її).
 2. Після відправки запиту показати анімацію (або повідомлення, [spinner](#)) про відправлення даних на місці схованих елементів або десь біля них.



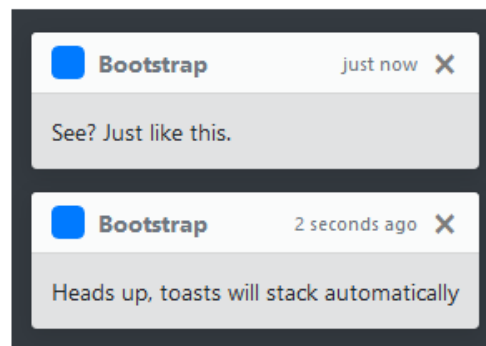
3. Після отримання відповіді від сервера сховати анімацію (повідомлення) процесу виконання запиту і виконати подальші дії
4. Якщо у відповіді прийшла помилка або запит не виконався успішно, показати повідомлення про помилку.

5. Веб сокет сервер:

1. Додати на веб сервер підтримку веб сокет сервера.
2. Коли створюється нова сутність на сервері, відправляти через веб сокет сервер повідомлення із даними цієї сутності.
3. Підключитись до веб сокет сервера на всіх сторінках вашого веб сайту.
4. Якщо на стороні браузера було отримано повідомлення про створення нової сутності, створити і показати графічне повідомлення (наприклад, [за допомогою цього](#)) у одному із кутів екрану.



- i. Повідомлення має містити посилання на сторінку нової сутності і кнопку для закриття повідомлення.
- ii. Повідомлення зникає через певний час або через дію користувача.
- iii. Повідомлення складаються у список ([stacking](#)), якщо в один момент часу їх декілька.



Зображення всіх розроблених компонент

Create university List of universities

Name

Country

Num of students

Campus

FoundationDate:

mm/dd/yyyy

University image:

Choose File No file chosen

Create

Last created (2):

University: kpi March 14th 2021, 11:14:19 pm

show more

University: rege March 14th 2021, 11:13:49 pm

show more

Create university List of universities

Name

Country

Num of students

Campus

FoundationDate:

mm/dd/yyyy

University image:

Choose File No file chosen

Info about university



ID: 604e7cabe26dc51900472033
Name: kpi
Country: Ukraine
Num of students: 12412
Campus: 3242342
Foundation date: 2021-03-18

Delete university

Last created (2):

University: kpi March 14th 2021, 11:14:19 pm

show more

University: rege March 14th 2021, 11:13:49 pm

show more

Create university

List of universities

Are you sure?

×

Confirm

Cancel

Name

Country

Num of students

Campus

FoundationDate:
mm/dd/yyyy

University image:
Choose File No file chosen

Create

Last created (2):

University: kpi March 14th 2021, 11:14:19 pm

show more

University: rege March 14th 2021, 11:13:49 pm

show more

Create university

List of universities

🔍 Enter the names of universities

You've entered: we

All entities (5):	
University: wefw March 14th 2021, 9:20:00 pm	show more
University: fwewe March 14th 2021, 9:20:17 pm	show more
University: wefw March 14th 2021, 9:47:52 pm	show more
University: qwe March 14th 2021, 9:50:16 pm	show more
University: wefwef March 14th 2021, 9:50:42 pm	show more

Висновок

Виконавши цю лабораторну роботу, ми навчились створювати динамічні веб-сторінки за допомогою браузерної мови JavaScript, попрацювали із технологією AJAX для отримання даних від REST API веб-сервера, а також використали WebSocket API для оповіщення сервером подій зміни даних.