Одеський коледж комп'ютерних технологій «Сервер»

Звіт про практичну роботу No 10

«API»

Виконав:

студент групи К21.1

Машкова Володимира

Перевірив:

Рудніченко Н.Д.

Одеса 2024

Вміст

[ВВЕДЕННЯ 2](#_Toc961076247)

[ТЕОРЕТИЧНА ЧАСТИНА 3](#_Toc1913604848)

[ПРАКТИЧНА ЧАСТИНА 4](#_Toc1900376073)

[ВИСНОВОК 9](#_Toc1091048837)

[ЛІТЕРАТУРА 9](#_Toc955073577)

# ВВЕДЕННЯ

Мета цієї лабораторної роботи - вивчення основних принципів створення API в Node.js з використанням шаблонів HBS та бібліотеки Bootstrap для стилізації сторінок. Ми розглянемо процес налаштування та створення API, а також інші ключові аспекти, які допоможуть нам краще зрозуміти, як працює Node.js з веб-технологіями.

Також ми вивчимо способи використання Express для створення HTML-сторінок та обробки введення/виведення за допомогою шаблонів HBS для більшої зручності. Для стилізації та покращення вигляду сторінок ми будемо використовувати бібліотеку Bootstrap.

# ТЕОРЕТИЧНА ЧАСТИНА

Node.js та Express - інструментарій для створення веб-додатків та API. Додатки, розроблені з використанням Node.js та Express, можуть мати широкий функціонал і ефективно взаємодіяти з користувачами.

У Node.js з використанням Express можна легко створювати маршрути та обробляти HTTP-запити. Використання шаблонів HBS дозволяє зручно форматувати відображення даних та створювати красивий вигляд веб-сторінок. Також можна використовувати бібліотеку Bootstrap для швидкої та зручної стилізації інтерфейсу.

Express надає велику кількість вбудованих модулів та middleware для роботи з асинхронним програмуванням. Ці модулі допомагають у взаємодії з клієнтами та обробці запитів, роблячи розробку веб-додатків на Node.js з Express ще простішою та зручнішою.

# ПРАКТИЧНА ЧАСТИНА

Підключаємо пакет hbs для роботи з Handlebars engine у нашій роботі (див. мал. 1).

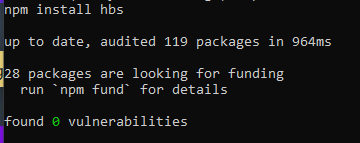


Рисунок 1 - Підключаємо пакет hbs.

Користувач має простий інтерфейс для заповнення списку тварин через API на Express, з простим функціоналом додавання, видалення та зміни (див. мал. 2).

{{> title}}

{{>formtitle}}

{{>footer}}

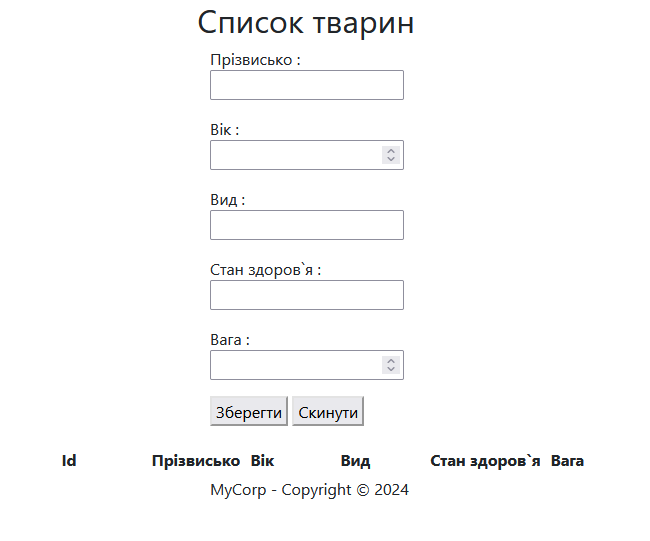


Рисунок 2 - Простий інтерфейс.

Додамо папугу під ім'ям Абд, який зараз, на жаль, хворіє (див. мал. 3).

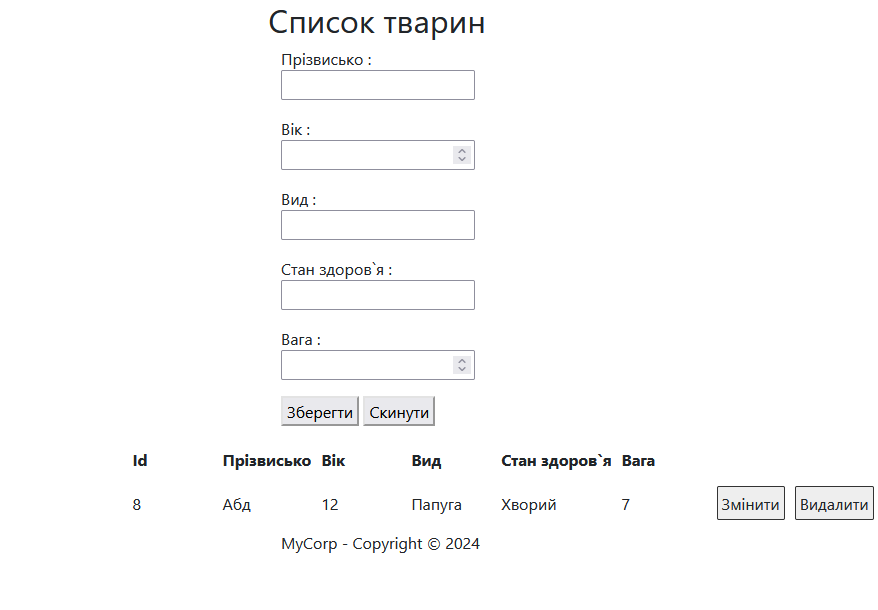


Рисунок 3 - Додача папуги.

Відредагуємо нашу папугу за допомогою кнопки "Змінити", підвищимо йому вік і вкажемо, що він тепер здоровий (див. мал. 4). Переглянемо наші зміни і побачимо, що всі його поля тепер оновлені (див. мал. 5).

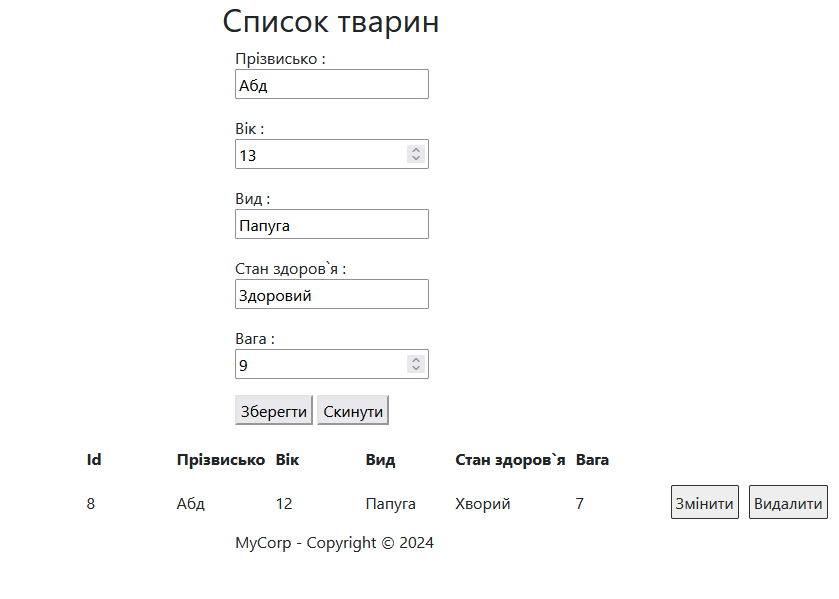


Рисунок 4 - Змінити поля папуги.

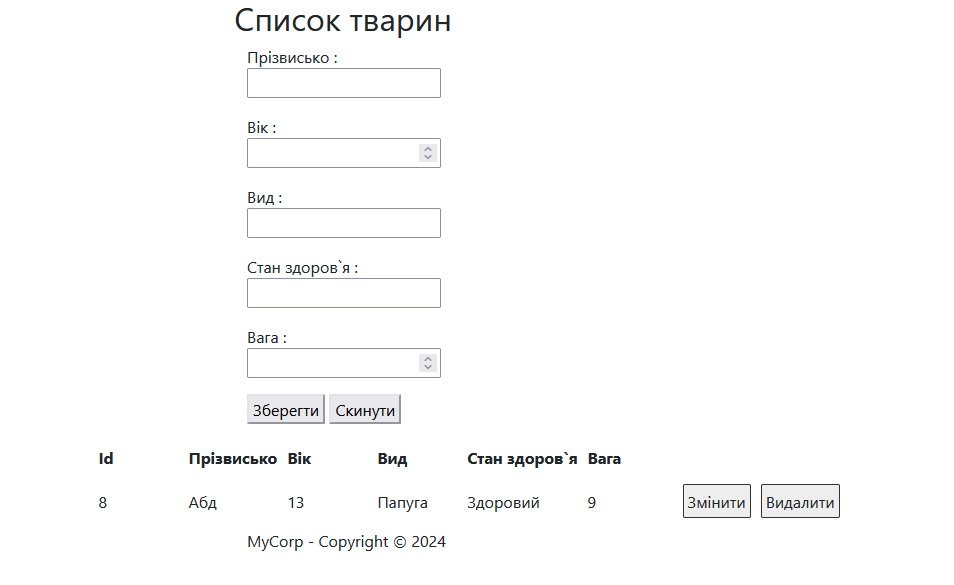
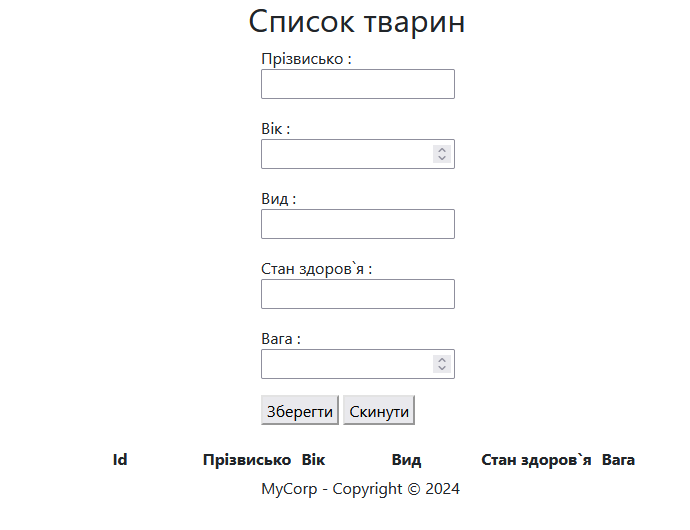


Рисунок 5 - Переглянемо зміни.

На останок, видалимо нашу папугу за допомогою кнопки "Видалити" і оновимо зміст списку (див. мал. 6).

Рисунок 6 - Видалимо папугу.

Встановимо пакет Bootstrap за допомогою npm install bootstrap, а потім підключимо його до нашого коду для стилізації (див. мал. 7).

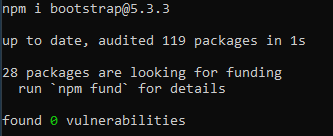


Рисунок 7 - Встановимо пакет Bootstrap

<link rel="stylesheet" href="node\_modules/bootstrap/dist/css/bootstrap.css">

<script src="node\_modules/bootstrap/dist/js/bootstrap.bundle.js"> </script>

# ВИСНОВОК

Під час цієї практичної роботи я зосередився на вивченні основних принципів роботи з API у Node.js з використанням фреймворка Express. Це дозволило мені отримати глибше розуміння того, як працювати з API на платформі Node.js.

В результаті цієї роботи я покращив свої навички роботи з використанням Express. Ця практична робота суттєво збільшила мою ефективність у створенні на Node.js.

# ЛІТЕРАТУРА

Відсутня