**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Інститут прикладної математики і фундаментальних наук**

**Кафедра прикладної математики**

**ЗВІТ**

**про виконання лабораторної роботи №4(6)**

**з курсу**

“**Програмування web-додатків ч.2”**

**СУБД MongoDB**

Виконав: студент групи ПМ-32

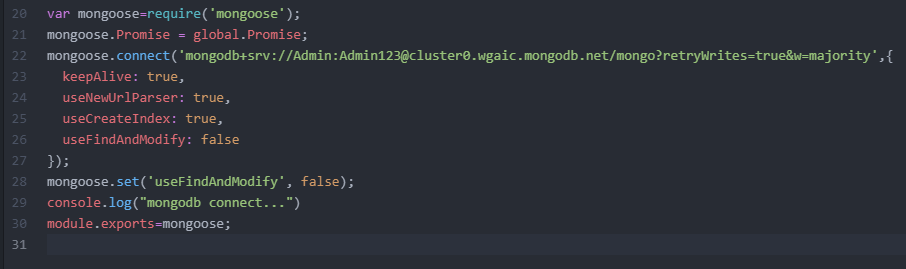
Михасів О.М.

Прийняв::

Пабирівський В.В.

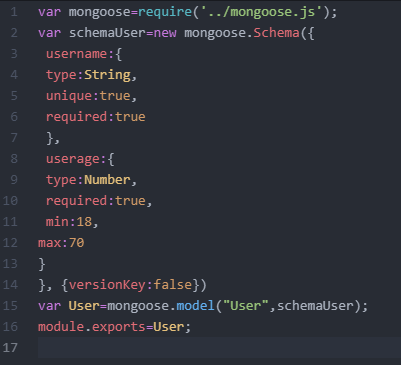
***Львів-2021***

**Постановка задачі:**

1. Реєструємось на сервісі https://mlab.com , який надає 500Мб безкоштовного хостінгу для нереляційних об'єктно-орієнтованих баз даних MongoDB.
2. Створюємо нову базу даних (Create New -> Sandbox Free (Plan type) ->Continue ->Europe (AWS Region) -> Continue -> mydb (Database Name) -> Continue -> Submit Order).
3. Заходим в створену базу даних mydb і додаєм для неї користувача (кнопка Add Database User на закладці Users). Для прикладу, введем наступні дані database username: Admin database password: Admin123 Дані створеного користувача будуть використовуватись при підключенні до бази даних з нашого проекту.
4. Інсталюєм ODM Mongoose для роботи з базою даних MongoDB npm install mongoose --save-dev
5. В корені проекту створюєм файл-модуль mongoose.js, в якому будемо встановлювати з'єднаня з базою даних   
   

В методі mongoose.connect задаєм стрічку підключення (connection string) до бази даних. Дану стрічку копіюємо з бази даних (To connect using a driver via the standard MongoDB URI) і змінюєм в ній параметри та на реальні дані створеного раніше користувача.

1. Спроектуємо модель User, яка буде створювати в базі даних колекцію users і взаємодіяти з нею (вибірка, додавання, знищення та оновлення даних). В корені проекту створимо каталог models з файлом user.js

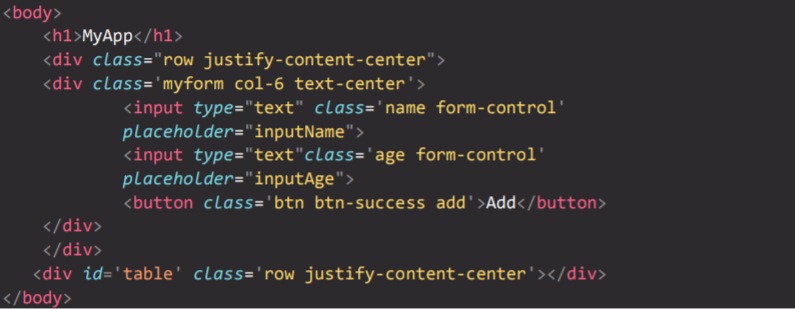


Дана модель буде здійснювати контроль типів та накладатиме обмеження на значення властивостей об'єктів з даними, які можуть міститись в колекції users.

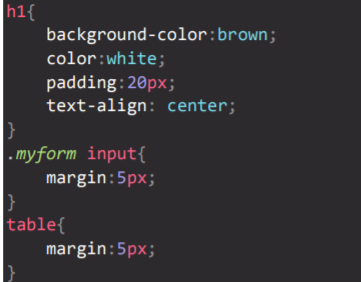
1. Підключаєм модуль-модель User в файлі server.js
2. Запускаєм сервер (файл server.js При усішному запуску сервера (відсутні помилки) в консоль Command Prompt виведеться повідомлення про з'єднання з базою даних, в якій створиться колекція usrers. Назва колекціїце назва моделі в множині (User -> users).
3. В файлі index.html під'єднуєм bootstrap.css перед стильовим файлом style.css



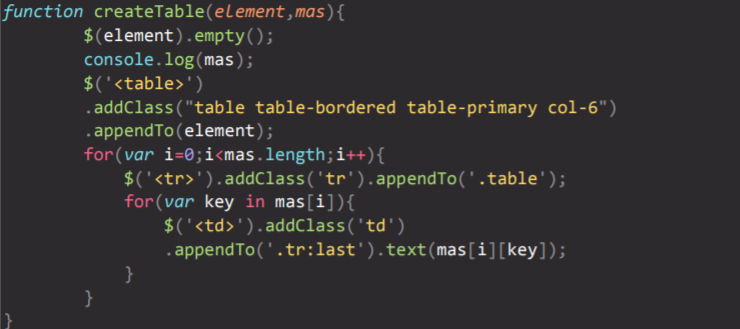
1. В файлі index.html створюєм форму для додавання користувачів в базу даних і додаєм bootstrap-класи для елемента div з id='table'



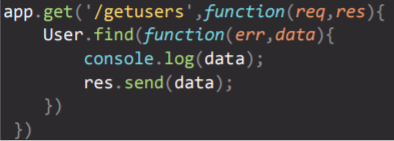
1. Додаємо стилі в файл style.css



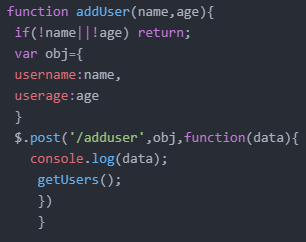
1. В файлі client.js додаєм bootstrap-класи table, table-bordered, table-primary та col-6 в функцію створення таблиці createTable(element,mas)



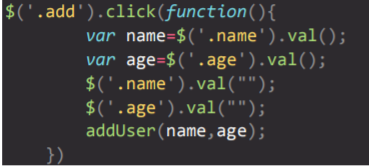
1. Модифікуємо обробник '/getusers' на сервері. На моделі User викликаємо метод find, який вибиратиме всі дані з колекції users. В параметр data отримуєм результат вибірки



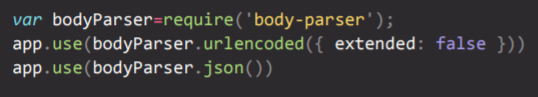
1. В файлі client.js створюєм функцію addUser(name,age), яка приймає параметрами ім'я користувача та його вік. Дана функція відправлятиме post-запитом на сервер свої параметри у вигляді об'єкта



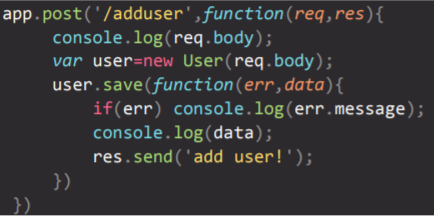
1. Програмуємо подію click для кнопки Add, яка має клас add



1. Інсталюєм модуль body-parser для post-запитів
2. Підключаєм модуль body-parser в файлі server.js і прив'язуємо його до express-проекту

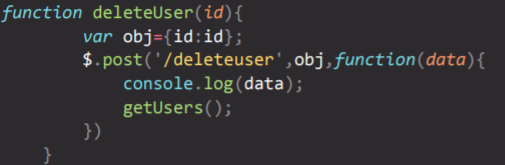


1. В файлі server.js створюєм обробник клієнтського запиту '/adduser', який буде додавати нового користувача в базу даних. Об'єкт з даними від клієнта буде доступний в req.body



При записі в базу даних об'єкт отримує унікальну властивість-ідентифікатор (\_id). Зверніть увагу, що метод save викликається на екземплярі моделі. Записаний в базу даних об'єкт повертається в параметр data

1. В файлі client.js створюєм функцію deleteUser(id), яка буде передавати на сервер унікальний \_id користувача

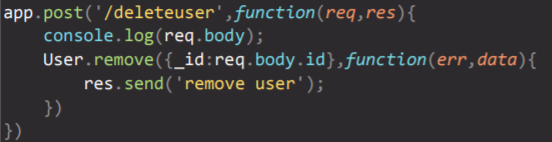


1. Приховуєм програмно відображення першого стовпчика таблиці з властивістю \_id. Також програмно додаєм в кінець таблиці стовчик, який міститиме в кожному рядку кнопку Delete для видалення користувача з бази даних. В функції createTable на ітерації зовнішнього циклу після внутрішнього циклу додаєм наступний фрагмент коду

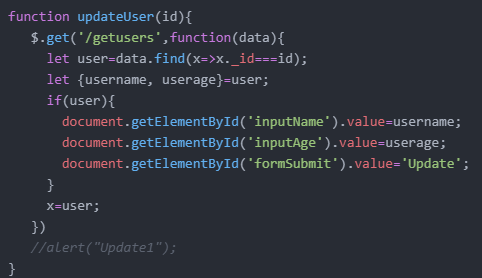


При кліку по кнопці Delete будемо зчитувати значення \_id в прихованому стовпчику відповідного рядка і викликати написану раніше функцію deleteUser.

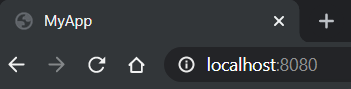
1. В файлі server.js створюєм обробник клієнтського запиту '/deleteuser', який буде видаляти користувача по отриманому значенні \_id.

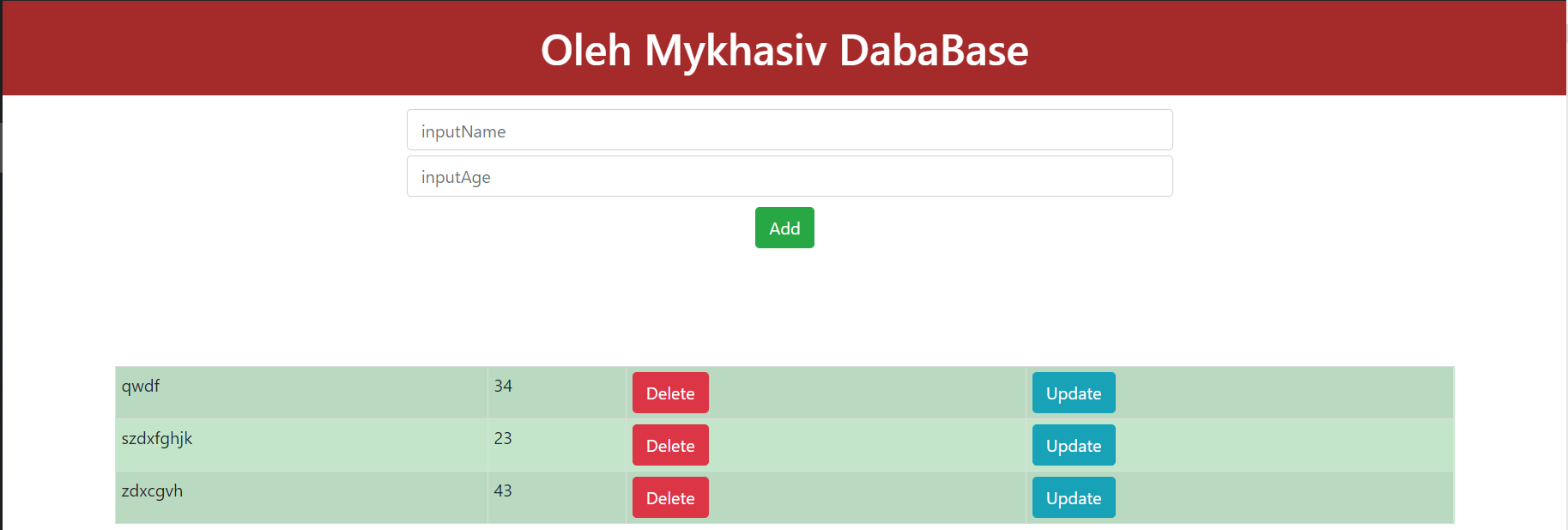


1. Реалізував оновлення існуючих даних в базі даних, в кожен рядок таблиці програмно додав кнопку Update.



**Результат:**

****

****

**Хід роботи:**

**Висновки:** у ході виконання роботи я ознайомився СУБД MongoDB, прив'язав базу даних до програми та покращив навички програмування в JavaScript.

**GitHub:**  <https://github.com/OlegMar1/WEB2/tree/main/Lab4>