**Лабораторна робота 17.**

**UI із використанням REST-API та реалізація операцій додавання, зміни та видалення даних.**

**Мета роботи:** Використовуючи bootstrap та jQuery реалізувати користувацький інтерфейс, що використовує розроблений раніше back-end rest-api. Доповнити функціонал back-end та front-end компонент реалізацією операцій додавання, зміни та видалення даних.

**Задачі:**

* Виконати реалізацією макету користувацького інтерфейсу із використанням bootstrap та jQuery
* Реалізувати AJAX запит на отримання даних від back-end сервісу та обробити отриманий результат
* Доповнити функціонал застосунку реалізацією операцій додавання, зміни та видалення даних
* Виконання індивідуального завдання згідно варіанту

1. Видалимо у корньовій папці нашого проекту index.php та створимо замість нього index.html. Також створимо папку «assest», а у ній «css» та «js». У «assest/css» скопіюємо «bootstrap.min.css» та створимо «main.js», а у «assest/js» скопіюємо «jquery-3.3.1.min.js» «popper.min.js», «bootstrap.min.js» та створимо «main.js». Підключимо всі css та js файли в index.html (рис.1)

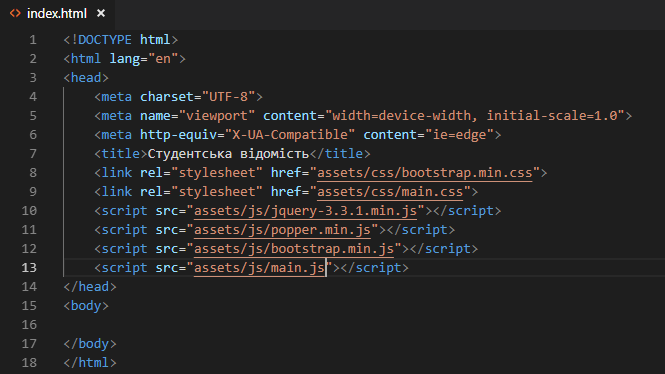


Рис.1. Початковий вигляд index.html

1. Реалізуємо get запит до нашого сервісу, розробленого у попередній роботі. Для цього додамо наступний код до нашого main.js (рис.2)

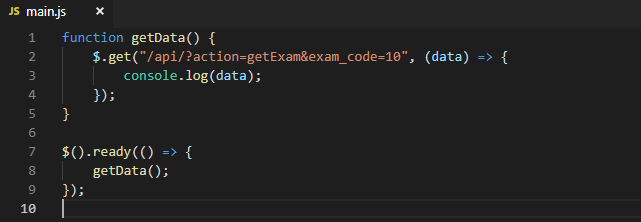


Рис.2. Get запит на отримання даних студентської відомості

Перевіримо результат, набравши у адресному рядку браузера «http://sql-server/» - поки повертається порожня сторінка (рис.3), але відкривши у браузері у режимі розробника (F12 у Chrome) вкладку консолі (рис.4) бачимо об’єкт «exam», що повернув наш бек-енд сервіс у форматі JSON.

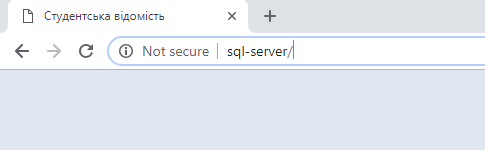


Рис.3. Головна сторінка

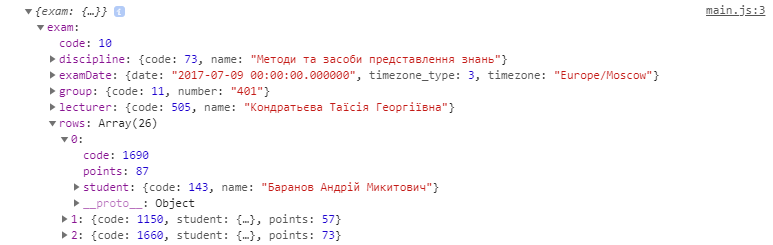
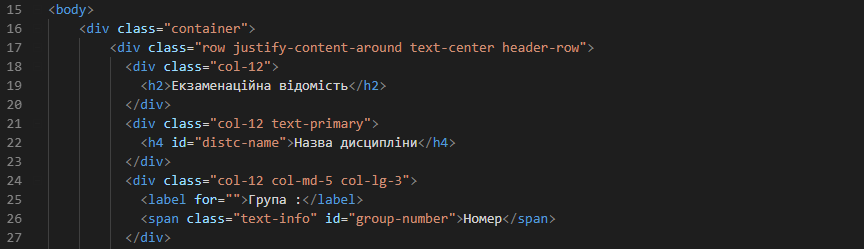
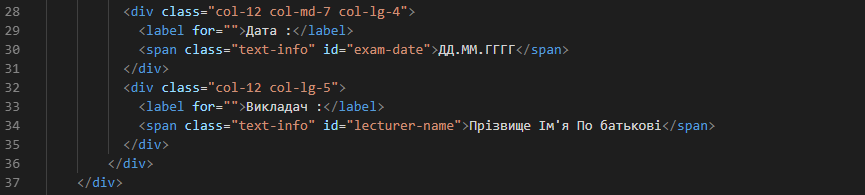
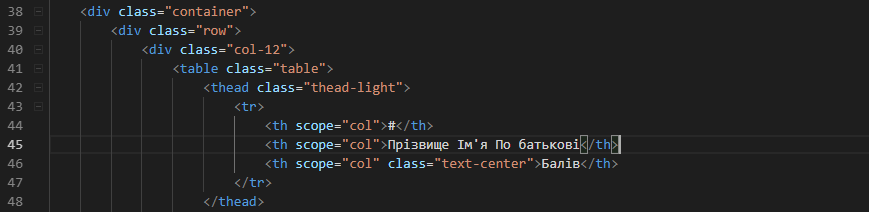


Рис.4. Об’єкт «exam» у форматі JSON, виведений на консоль

1. Створимо шаблон для UI у index.html (рис.5) та додамо декілька стилів до main.css (рис.6)







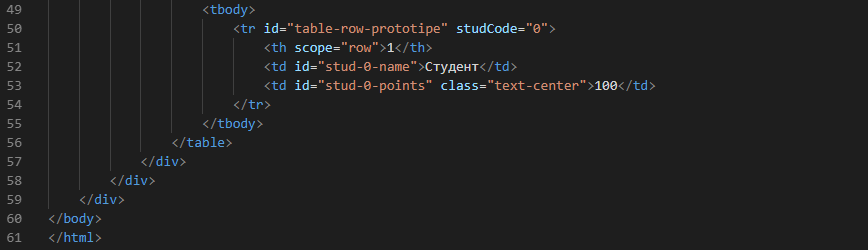


Рис.5. Шаблон для UI у index.html

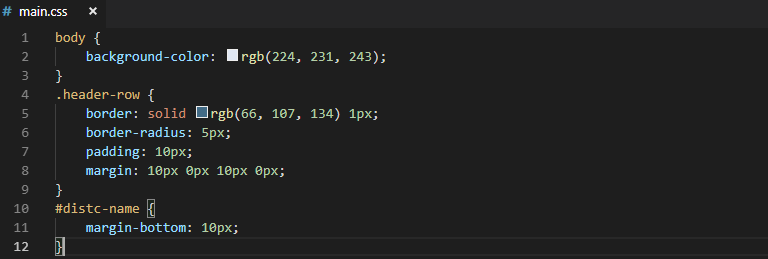


Рис.6. Вміст файлу main.css

Переглянемо результат (рис.7)

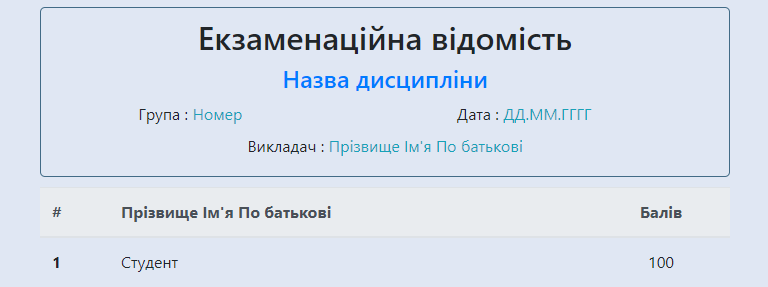
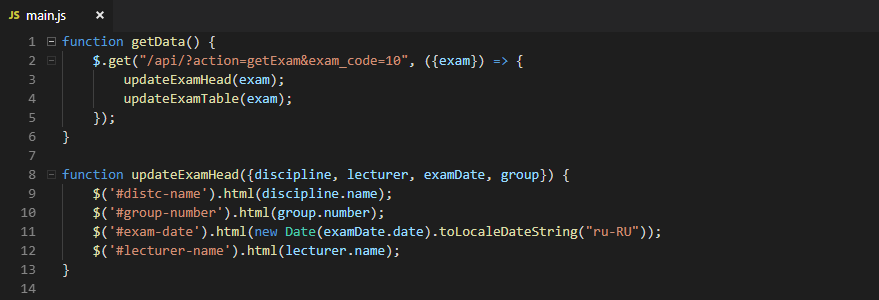
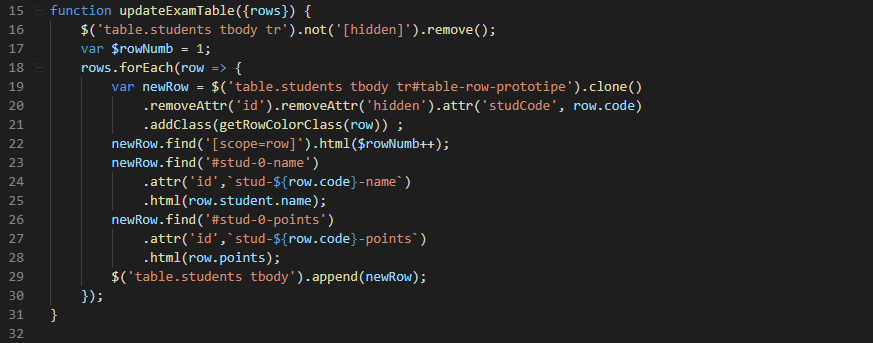


Рис.7. Зовнішній вигляд шаблону відомості

1. Виконаємо заповнення шаблону даними після їх отримання в main.js (рис.8)





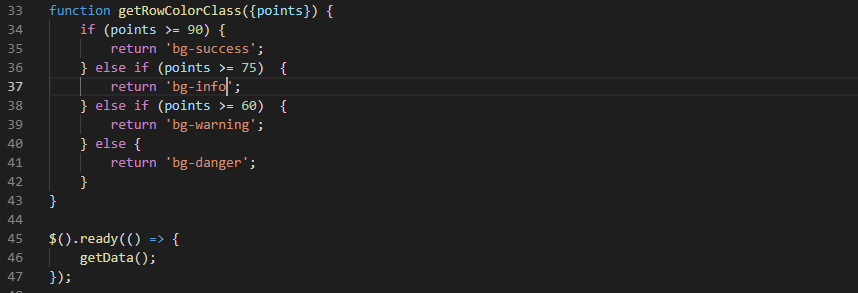


Рис.8. Оновлення даних на сторінці

Переглянемо результат роботи на головній сторінці (рис.9)

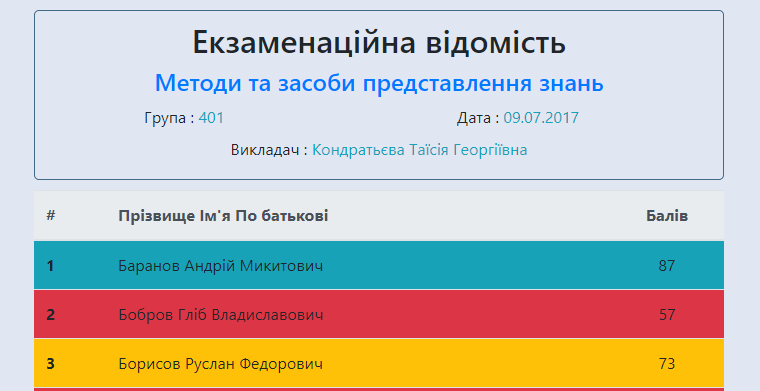
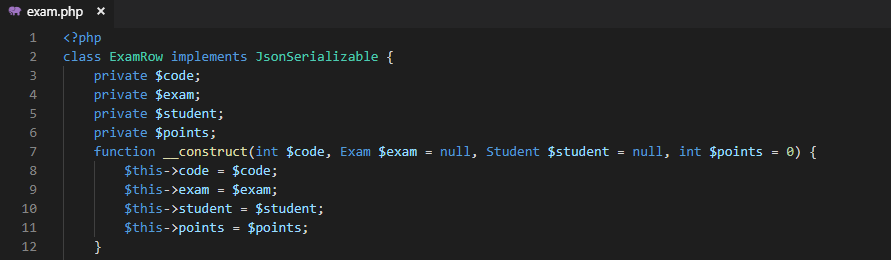
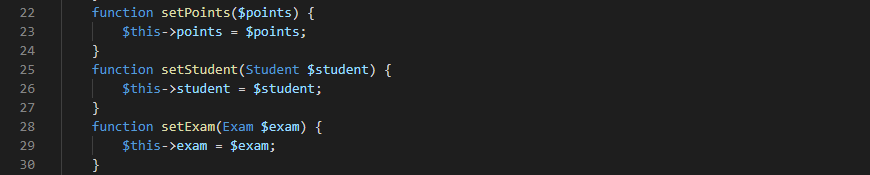


Рис.9. Зовнішній вигляд відомості після оновлення даних

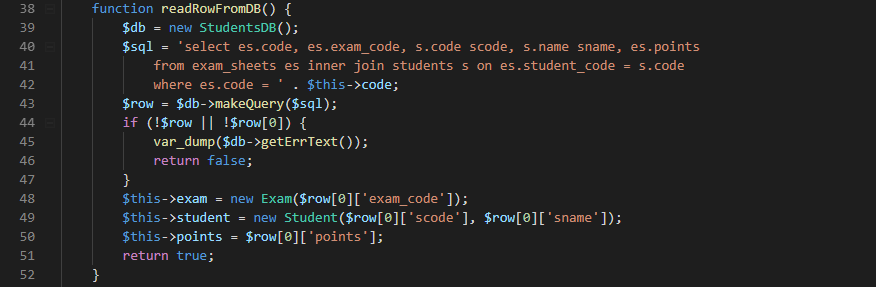
1. Забезпечимо редагування даних нашої відомості. Для цього додамо до нашого back-end rest-API функції додавання, зміни та видалення рядку із нашої відомості. Почнемо із редагування класу «ExamRow» у файлі «data/exam.php» (рис.10). Тут додаються гетери, сетери та методи для взаємодії із базою даних при виконанні операцій читання, додавання, зміни та видалення рідку.

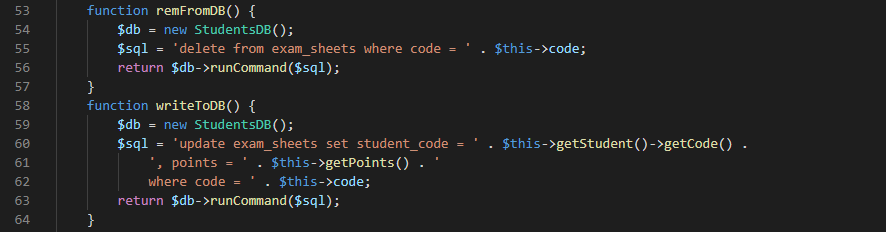












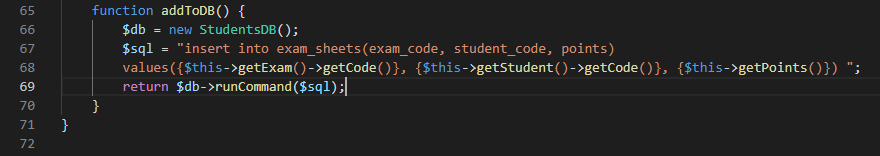
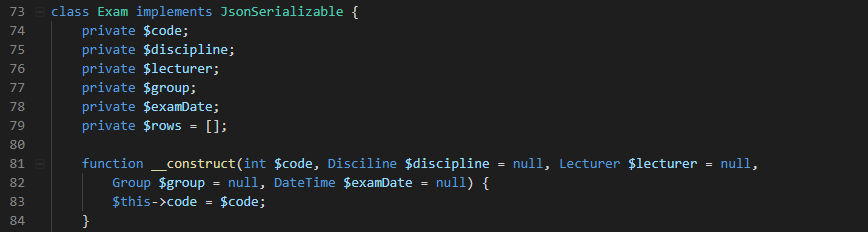
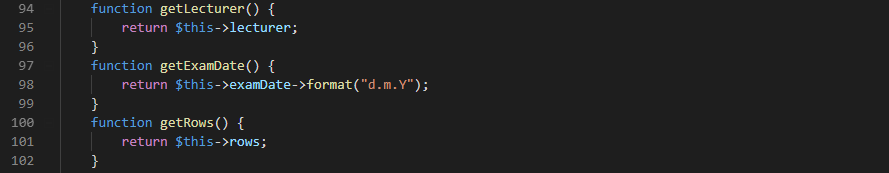


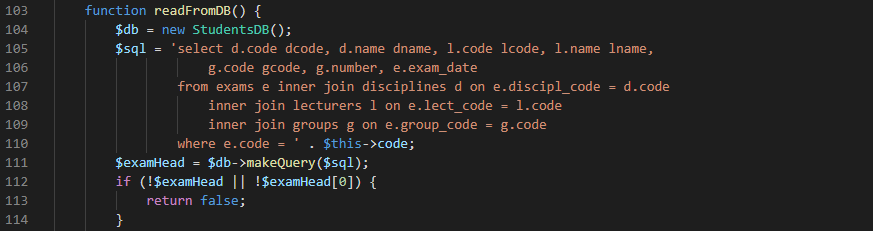
Рис.10. Оновлений клас «ExamRow»

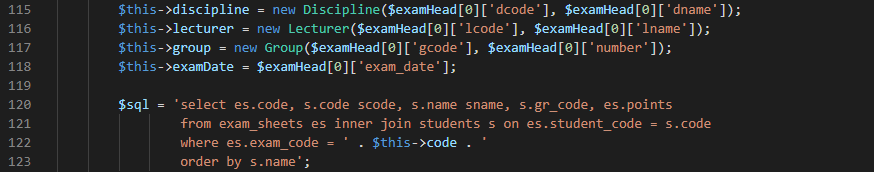
Далі виконаємо необхідні зміни для класу «Exam». Тут також додаються необхідні гетери та сетери, а також метод, що повертає список студентів групи, яка вибрана у відомості, але яких у відомості ще немає (кого можна додати до списку) (рис.11)

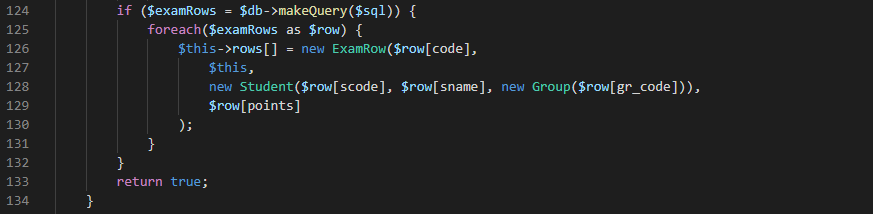


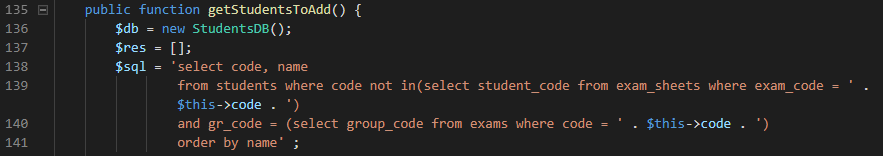


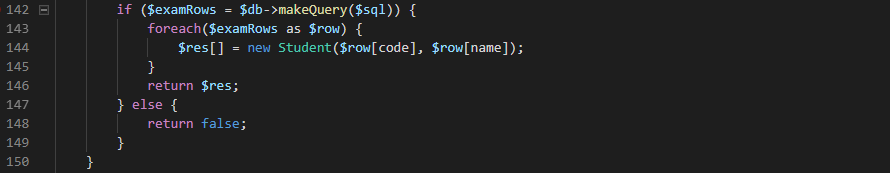












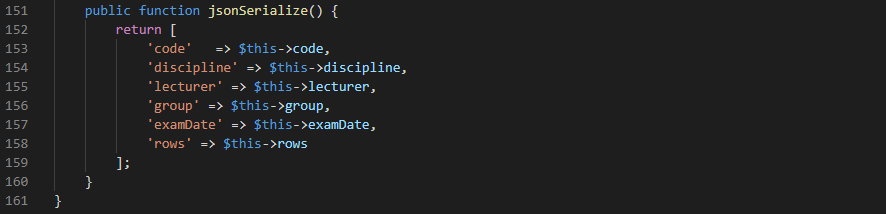
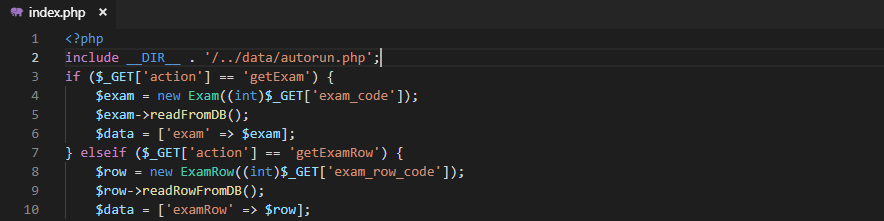
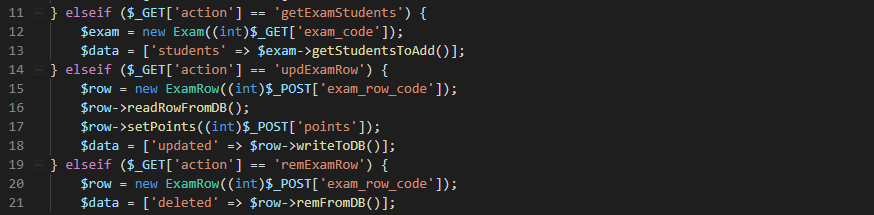


Рис.11. Оновлений клас «Exam»

Змінимо наш сервіс (файл api/index.php) для того, щоб він обробляв команди отримання даних рядку, списку студентів на додавання, та команди додавання, зміни та видалення рядку (рис.12)





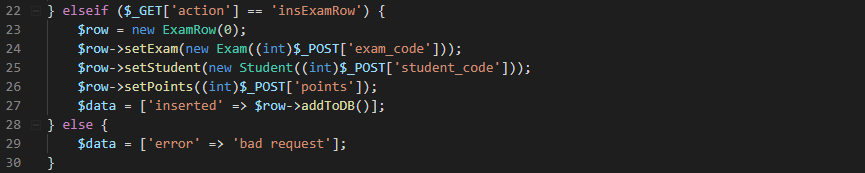
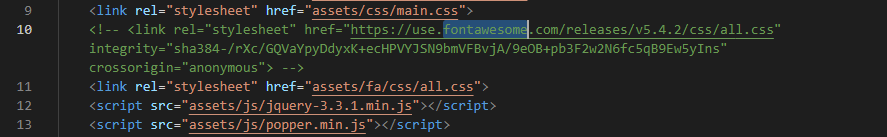
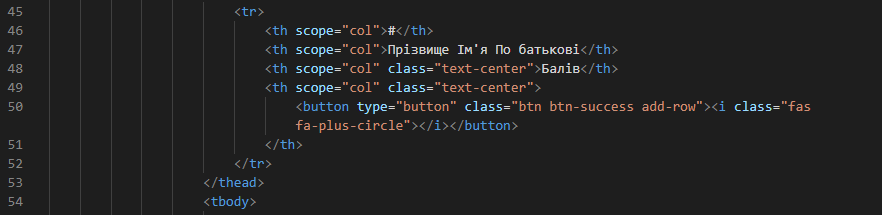


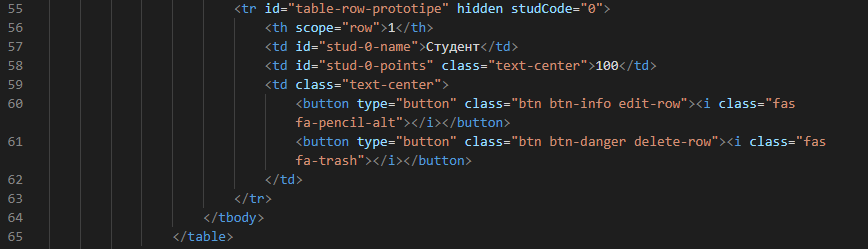


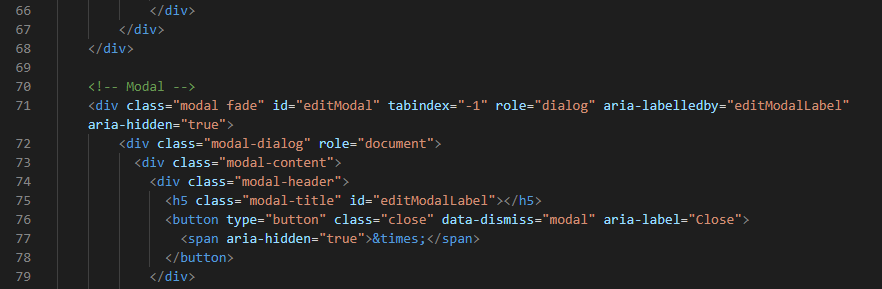
Рис.12. Оновлений rest-API сервіс

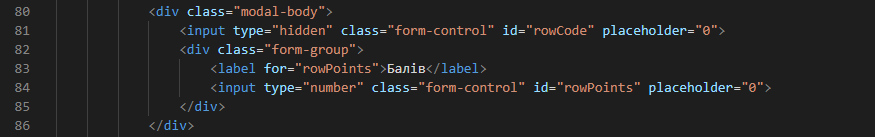
1. Тепер змінимо UI для можливості взаємодії із новими функціями back-end та відображення результату їх виконання на сторінці. Почнемо із index.html (рис.13). Тут підключаємо «fontawesome» для можливості відображення на кнопках цікавих іконок (підключити можна як за допомогою CDN, так і завантажити локально), додаємо на сторінку кнопки для додавання, зміни та видалення рядку, та додаємо також два модальних вікна – для додавання та зміни рядку нашої відомості.

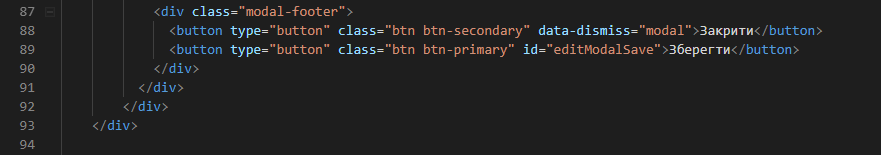


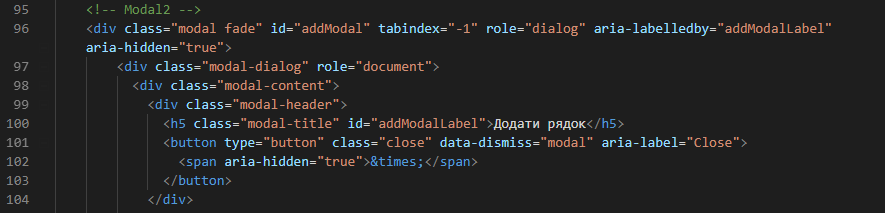


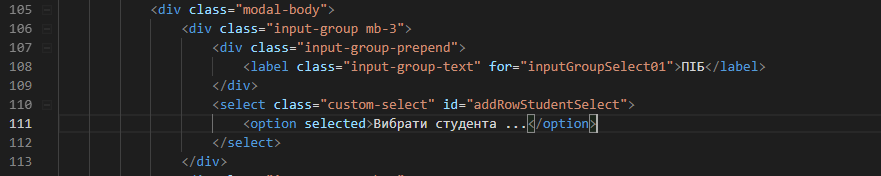


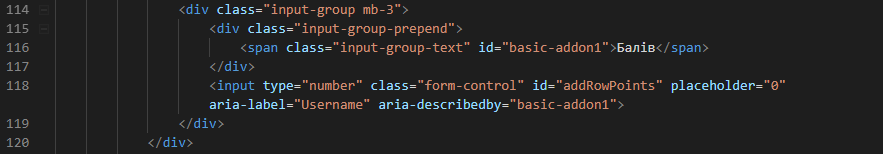












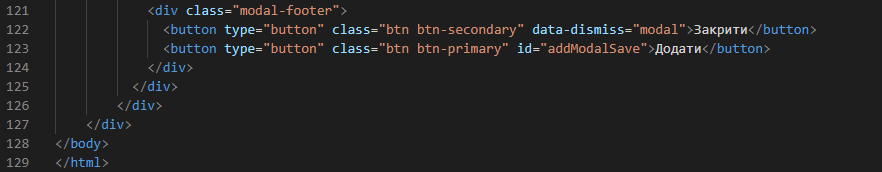
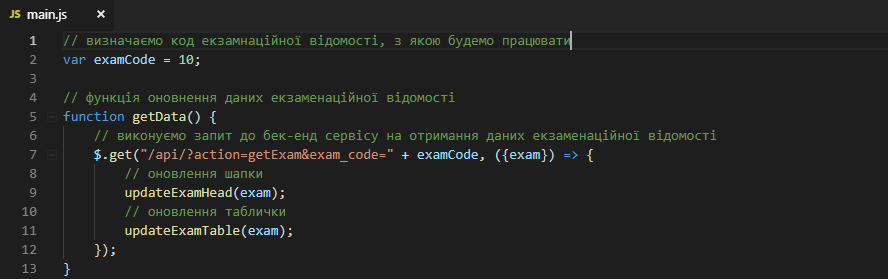
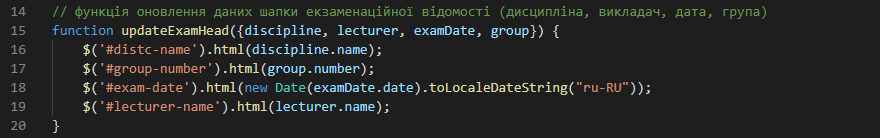
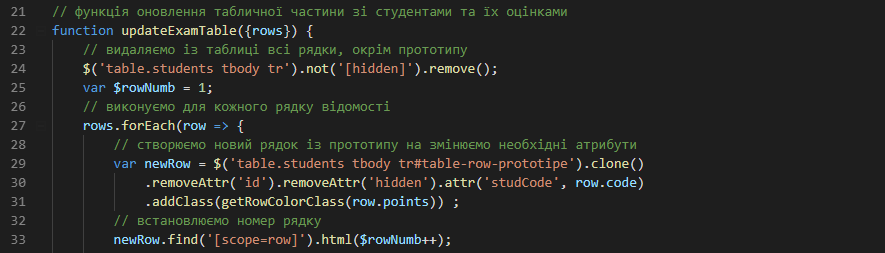


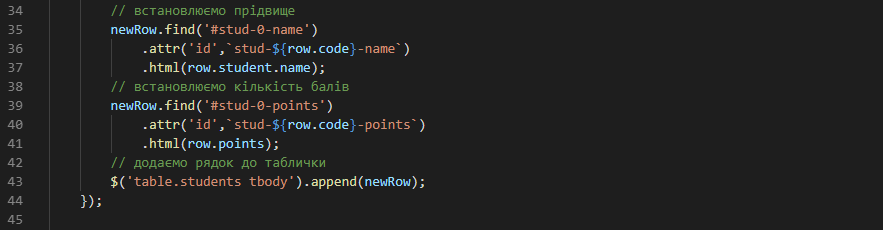
Рис.13. Змінений index.html

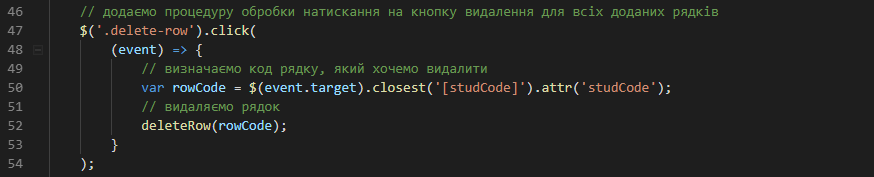
Виконаємо обробку натискання всіх доданих в UI кнопок у main.js (рис.14)

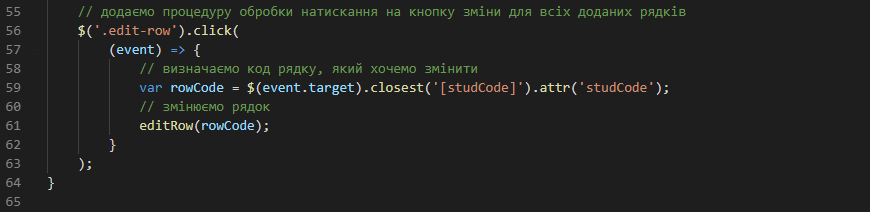


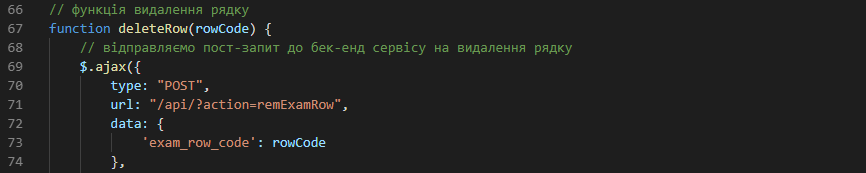


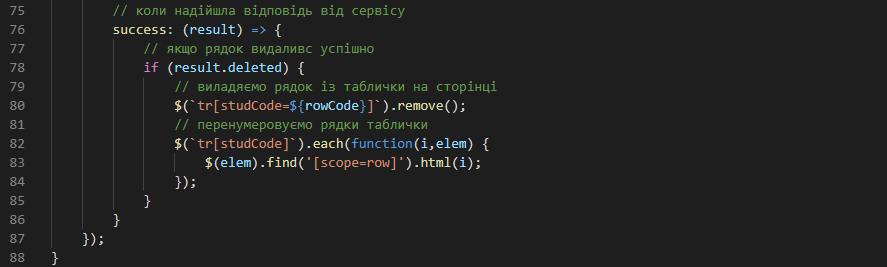


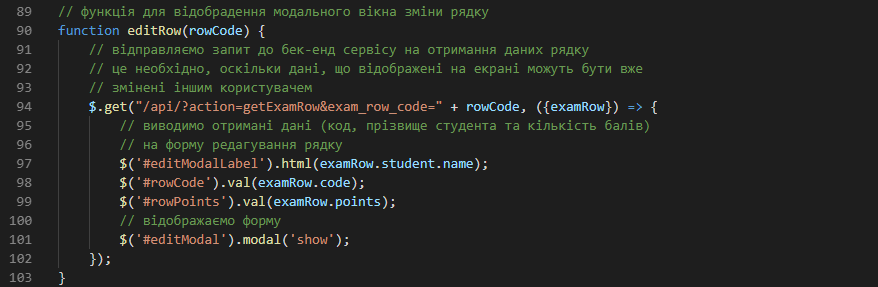


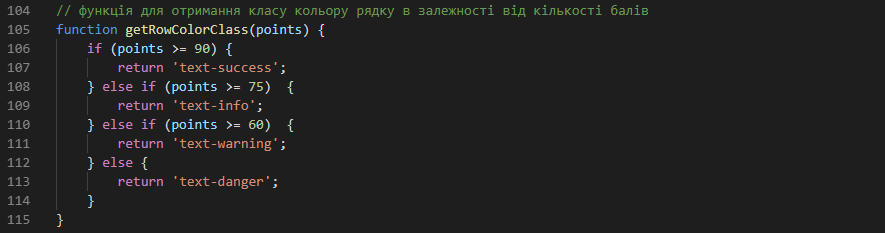


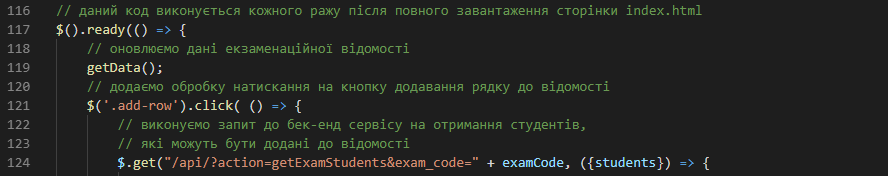


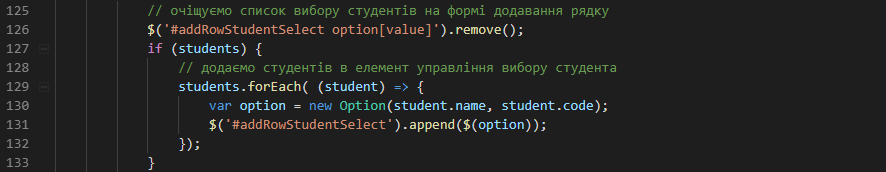


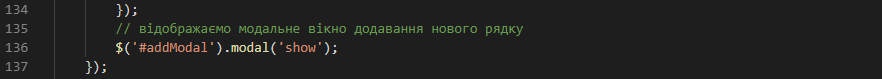


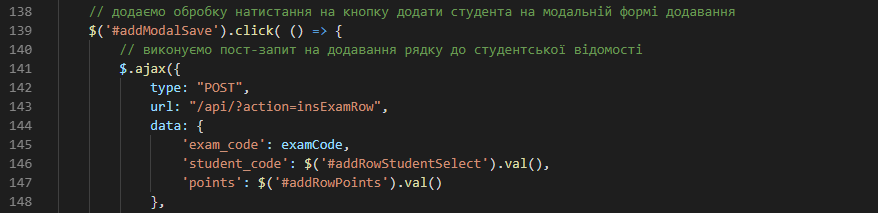


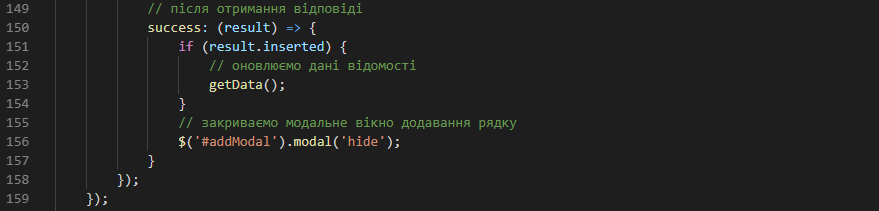


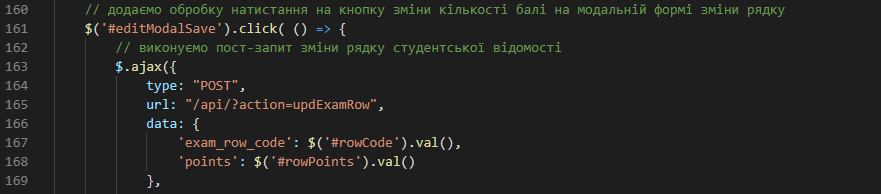












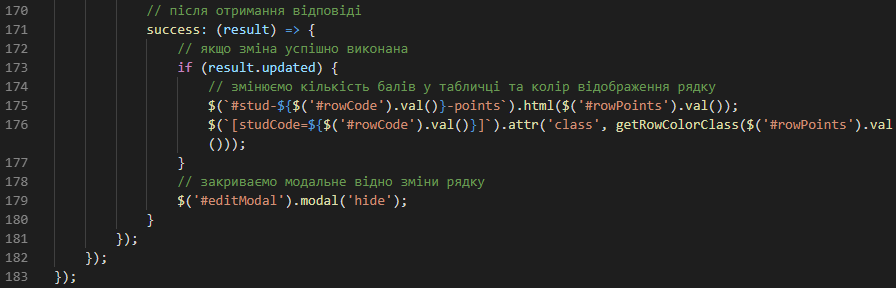


Рис.14. Змінений main.js

Перевіримо роботу нашого застосунку. Виконаємо видалення, зміну та додавання студенту у відомість (рис.15-17)

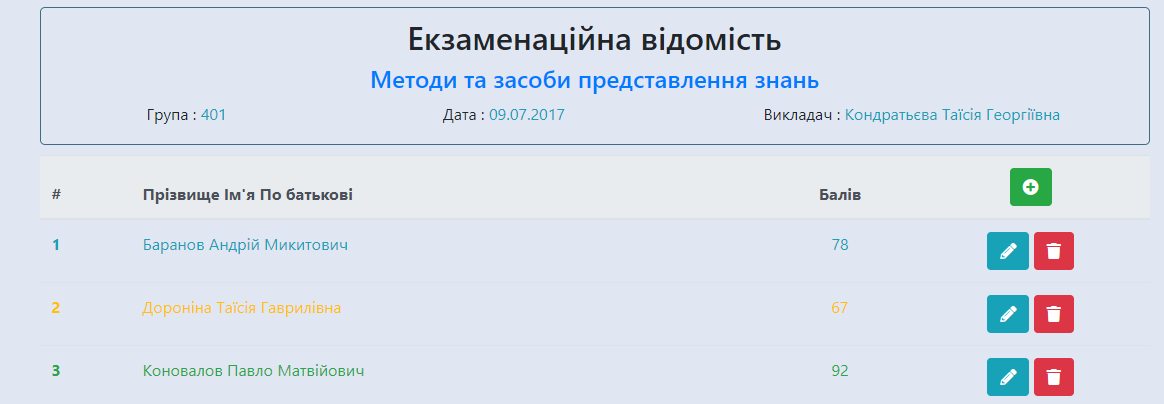


Рис.15. Вигляд сторінки після змін

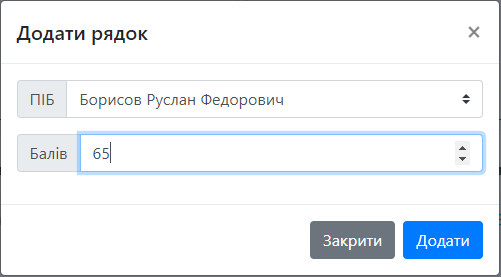


Рис.16. Форма редагування рядку студентської відомості

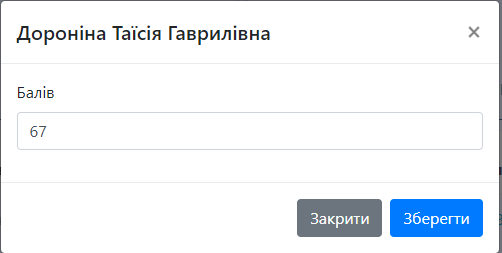


Рис.17. Форма додавання нового рядку студентської відомості

1. Кінець роботи

**Завдання для індивідуального виконання.**

Для БД свого варіанту та таблички, обраної при виконанні індивідуального завдання попередньої роботи, реалізувати користувацький інтерфейс, що взаємодіє із реалізованим у попередній роботі back-end сервісом.

У back-end та front-end реалізувати виконання операцій додавання, зміни та видалення рядку таблички.