

## Обмін даними у вебзастосунках

Можна виконувати під яку конкретну тему або на кожну із лабораторних обирати нову предметну область. Розробка програми в одній предметній області буде оцінюватись вище.

Дозволяється працювати в командах - до 6 людей. При роботі в команді дозволяється тільки проєкт в межах однієї предметної області, тобто в одному застосунку мають бути реалізовані усі види комунікації. Плюси в тому, що кожен з людей може зайнятись окремим видом комунікації, або один бере написання клієнтів, а інший серверів. Також дозволяється мати в команді людей із ролями Бізнес Аналітик, Проджект Менеджер, до яких будуть окремі вимоги - тобто бізнес аналітик із своєї сторони має розписати документацію застосунку, діаграми того як має працювати програма і якість буде оцінювати зрозумілістю при читанні і покриттю більшості нюансів. Проджект Менеджер має ж звітуватись про процес виконання роботи, спостерігати, що застосунок пишеться під кожну лабораторну на якій проджект менеджер, або хтось із команди проводить демонстрація. Для тестувальників може підійти остання лабораторна робота, де потрібно переконатись, що все працює, зберегти усі тестові кейси для подальшої демонстрації (для успішної роботи розробникам треба залишити чіткі вказівки як можна запустити застосунок)

## Задачі

- Реалізувати сервер, який буде приймати REST HTTP запити - усі базові типи запитів: POST, PUT, GET, DELETE
- Реалізувати клієнт, який буде надсилати запити на сервер. Клієнт має мати "зручний" інтерфейс (на 3 і консоль підійде)

## Приклад реалізації із використанням мови C#

В даному прикладі можна знайти приклад реалізації REST сервера в двох екземплярах: [конролери](#) і [minimal api](#). Minimal API має також [приклад реалізації комунікації в режимі реального часу](#) у вигляді web sockets та SignalR.

Також є [приклад реалізації клієнту у вигляді консольного застосунку](#)

<https://gitlab.com/communication-in-web/rest-api>

Для більших деталей можна переглянути лекцію [2. HTTP/HTTPS. XML/JSON. REST API](#)