

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА
Кафедра комп'ютерних наук та інформаційних систем

Лабораторна робота № 2
з курсу
«Розробка WEB-додатків (TypeScript)»
на тему:
«Проектування веб-додатків TypeScript»

Виконав:

студент групи КН-41
Кадемський О. В.
«28» лютого 2025 р.

(підпис)

Перевірив:

ст. викладач кафедри КНІС
к.т.н. Ізмайлов А. В.
«__» _____ 2025р.

(оцінка, підпис)

Мета роботи: Навчитись проектувати веб-додатки TypeScript.

Теоретичні відомості:

TypeScript (TS) – це JavaScript (JS) із синтаксисом для типів. TS є строго типізованою мовою та є інструментом для розробників, який покликаний допомогти створювати якісніший JS-код.

TS компілюється у JS і фактично є розширенням (superset) JS. Процес компіляції TS полегшує та стабілізує процес JS-розробки, оскільки дозволяє виявляти та виправляти помилки, які, у загальному випадку, довелося б виявляти уручну при традиційному процесі розробки у JS.

Для встановлення TS слід встановити Node Package Manager (npm). Після цього можливе встановлення TS.

Глобальне встановлення TS: `npm i typescript -g`

Для швидкого тестування успішності встановлення TS слід:

- у порожній папці підготувати мінімальну html-сторінку до якої підключатиметься скрипт;
- створити файл для TS, наприклад `main.ts`;
- внести у `main.ts` довільний код, наприклад:

```
let username = 'John';  
console.log(username);
```

- у консолі (PowerShell, Bash, тощо) запустити команду TS (`tsc - TypeScript Command`): `tsc main.ts`

Хід роботи:

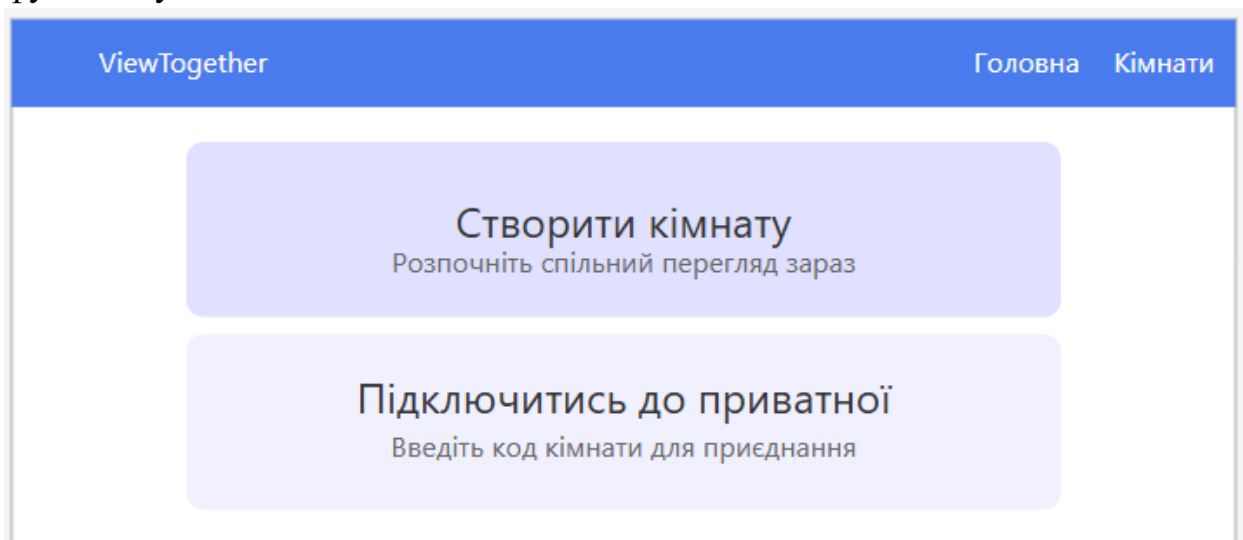
1.2.1 Сформулювати ідею деякого веб-додатку, створеного за допомогою TypeScript. Запропонований веб-додаток повинен забезпечувати взаємодію із серверним компонентом, взаємодію із користувачем, здійснювати маршрутизацію (routing) клієнтської частини, тощо. Описати сформульовану ідею у межах 5-10 речень:

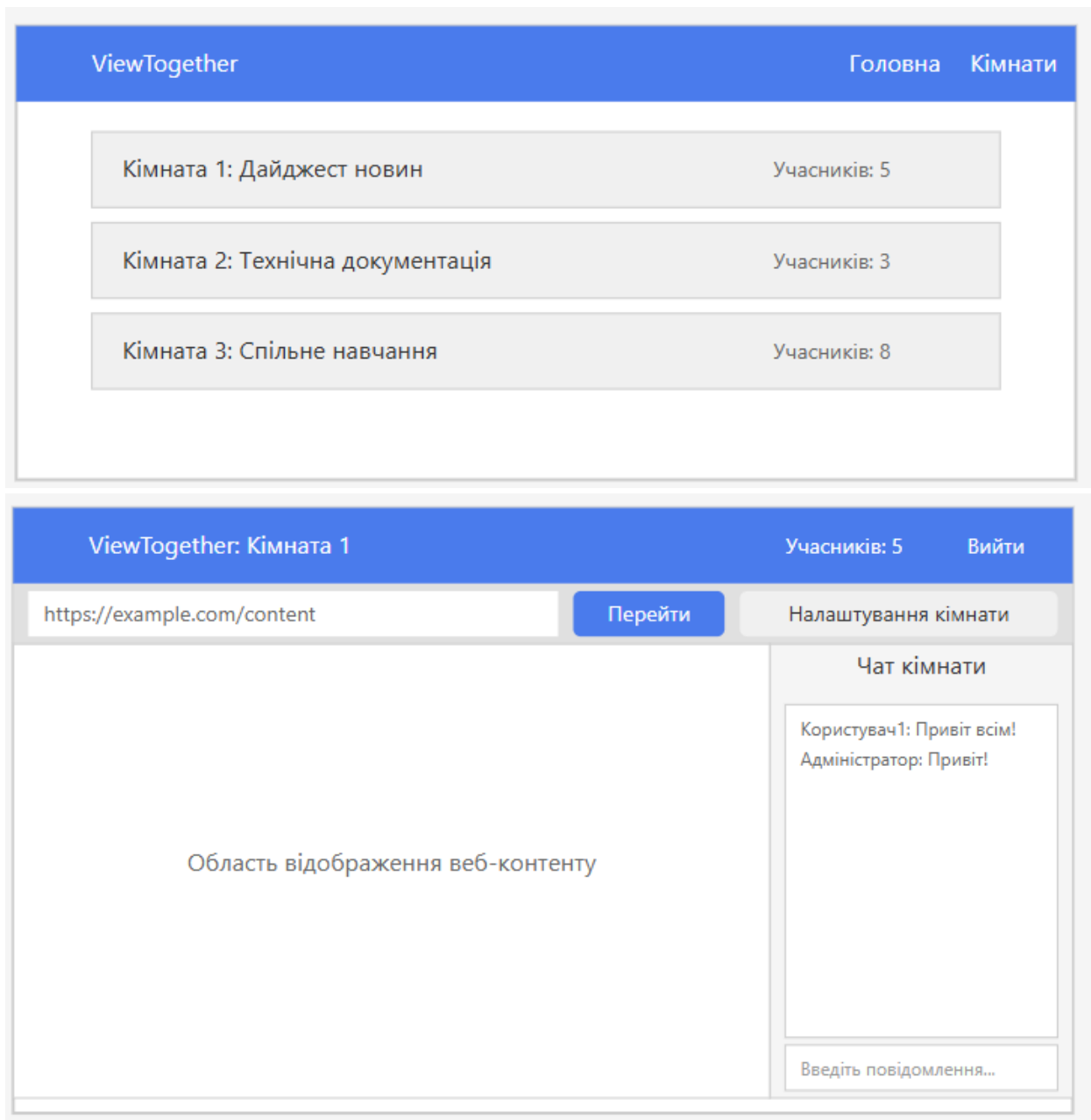
Веб-додаток "ViewTogether" - це платформа для спільного перегляду веб-контенту без необхідності реєстрації та авторизації. Додаток дозволяє користувачам створювати віртуальні кімнати, ділитися посиланнями на будь-які веб-сайти із відео та одночасно переглядати їх з іншими учасниками.

Кожна кімната має свого адміністратора, який встановлює правила перегляду (дозвіл на зміну контенту, можливість чату, т.п.). Платформа включає три основні сторінки: головну з можливістю створити кімнату, сторінку списку всіх публічних кімнат для приєднання та безпосередньо сторінку кімнати з інтерфейсом перегляду.

Унікальність додатку полягає в тому, що він не потребує бази даних, використовуючи технології веб-сокетів для прямої взаємодії учасників у режимі реального часу.

1.2.2 Навести (та включити до звіту) приклади макетів інтерфейсу користувача у запропонованому веб-додатку, наприклад за допомогою каркасних діаграм, інструментів для створення прототипів, та інших засобів побудови імітаційних представлень дизайну інтерфейсу користувача. Додати пояснення щодо призначення елементів інтерфейсу та їх функціонування:



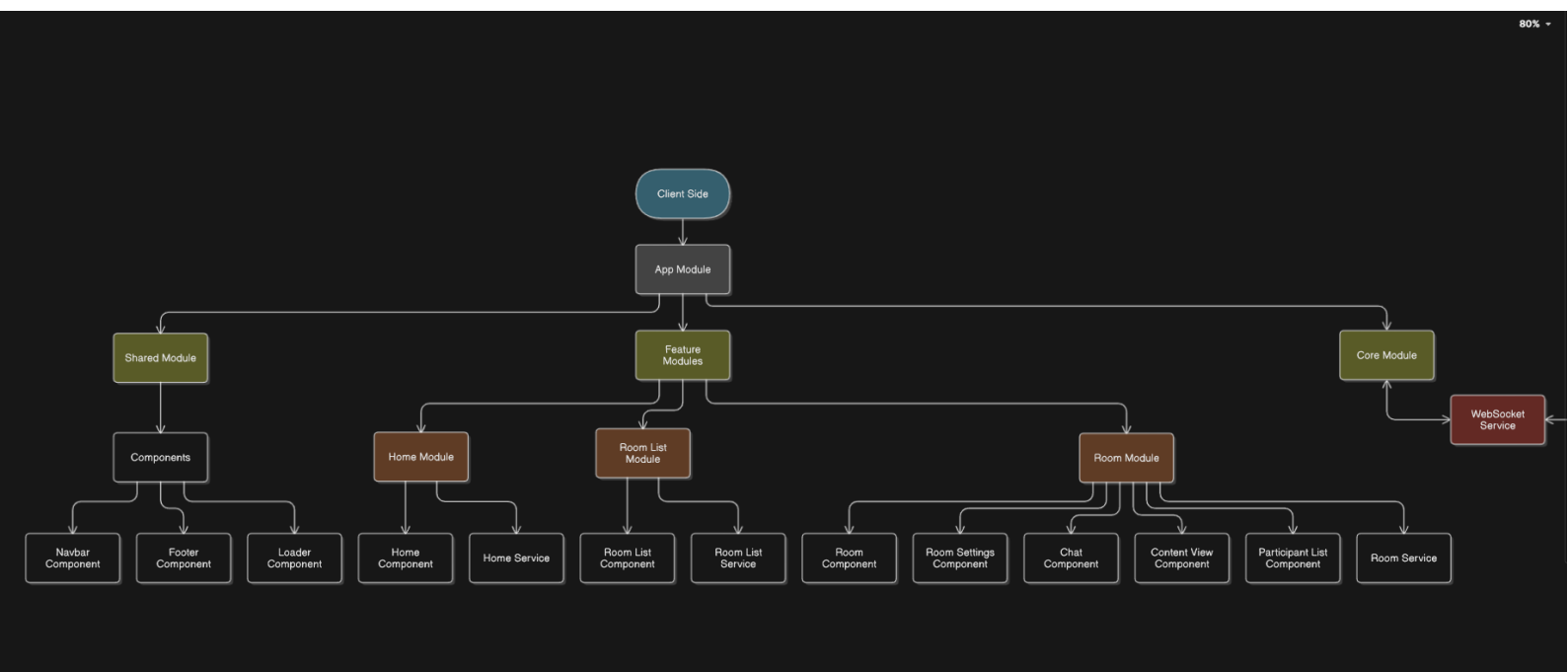


1.2.3 Розробити ієрархію (систему зв'язків) модулів веб-додатку. Навести її у вигляді схеми (діаграми) та включити до звіту пояснення до неї:

Клієнтська частина (Angular):

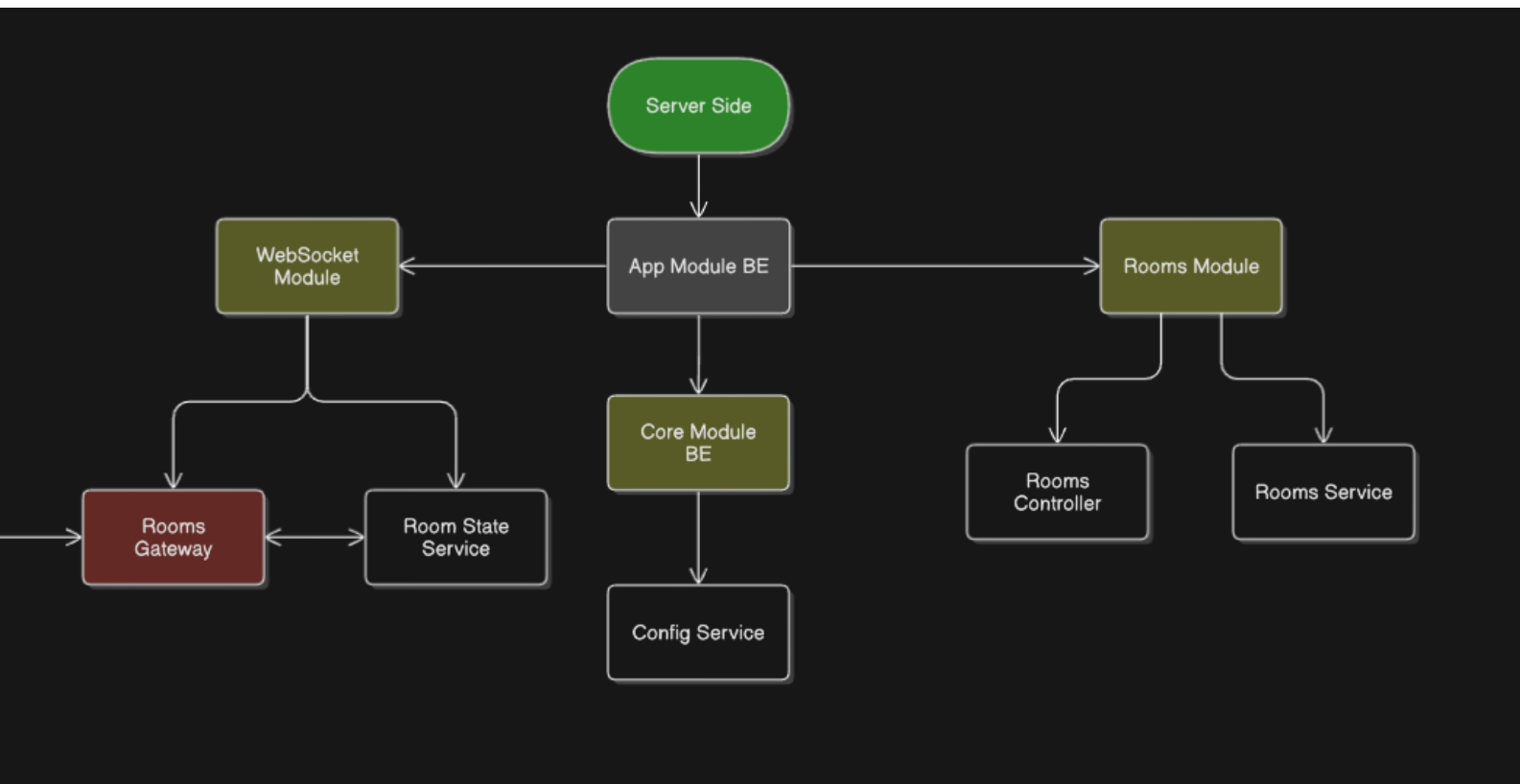
1. **AppModule** - головний модуль додатку, який об'єднує всі інші модулі.
2. **CoreModule** - містить сервіси, які використовуються в усьому додатку:
 - **WebSocketService** - забезпечує взаємодію через WebSocket
3. **SharedModule** - містить спільні компоненти:
 - **Components** – базові компоненти
4. **FeatureModules** - модулі окремих функціональних частин додатку:
 - **HomeModule** - модуль головної сторінки
 - **RoomListModule** - модуль списку кімнат

- **RoomModule** - модуль кімнати перегляду, який містить:
 - **RoomComponent** - головний компонент кімнати
 - **RoomSettingsComponent** - компонент налаштувань кімнати (для адміністратора)
 - **ChatComponent** - компонент чату кімнати
 - **ContentViewComponent** - компонент для відображення веб-контенту
 - **ParticipantListComponent** - компонент списку учасників
 - **RoomService** - сервіс для роботи з кімнатою



Серверна частина (NestJS):

1. **AppModule_BE** - головний модуль серверної частини.
2. **CoreModule_BE** - базові сервіси сервера:
 - **ConfigService** - сервіс конфігурації
3. **RoomsModule** - модуль для роботи з кімнатами:
 - **RoomsService** - сервіс з бізнес-логікою для кімнат
 - **RoomsGateway** - обробник WebSocket-подій для кімнат
4. **WebSocketModule** - модуль для роботи з WebSocket:
 - **WebSocketGateway** - основний шлюз WebSocket
 - **RoomStateService** - сервіс управління станом кімнат в реальному часі



1.2.4 Розробити діаграму класів веб-додатку та включити її до звіту разом із поясненнями до неї:

IRoom:

- додавання/видалення учасників
- оновлення URL та налаштувань
- перевірки прав користувачів
- інтеракції із користувачами та вебсокетами

IUser – інтерфейс користувача з методами для:

- зміни ролі

IRoomSettings – інтерфейс налаштувань з методами для:

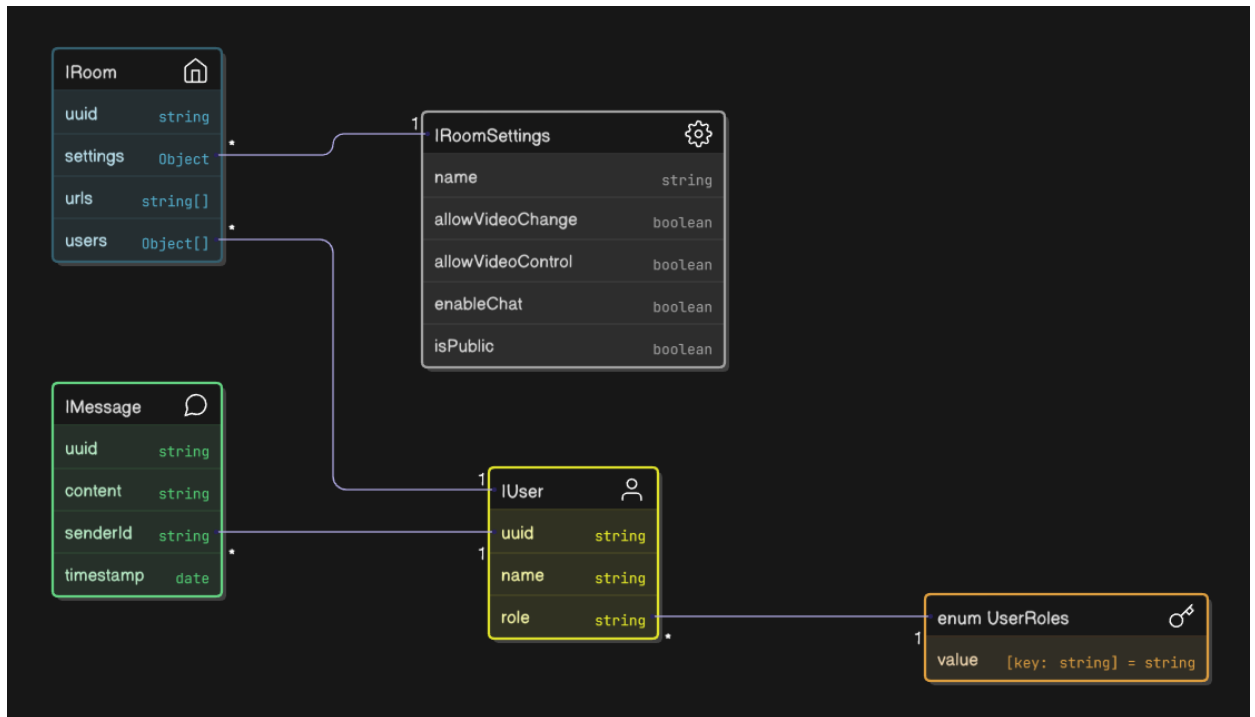
- перемикання різних опцій кімнати

IMessage - інтерфейс повідомлення для:

- відображення всім приєднаним користувач

UserRoles - еnumерація ролей користувачів:

- ADMIN - адміністратор
- USER - звичайний користувач



Висновки:

У рамках виконаної роботи було розроблено комплексну документацію для веб-додатку "ViewTogether" на базі TypeScript із використанням фреймворків Angular (клієнтська частина) та NestJS (серверна частина). Створений додаток відповідає всім заданим вимогам, забезпечуючи взаємодію із серверним компонентом, інтерактивний інтерфейс користувача та маршрутизацію на клієнтській стороні.

Основні результати виконаної роботи:

1. **Формулювання ідеї** - розроблено концепцію веб-додатку для спільного перегляду веб-контенту без авторизації та бази даних, з розподілом ролей та можливістю налаштування правил кімнати.
2. **Проектування інтерфейсу** - створено макети для трьох основних сторінок додатку (головна, список кімнат, кімната перегляду) з детальним описом елементів інтерфейсу та їх функціонального призначення.
3. **Розробка архітектури** - спроектовано ієрархію модулів додатку, що забезпечує чіткий розподіл відповідальності та гнучкість при розширенні функціоналу. Виділено основні модулі для клієнтської та серверної частин, а також встановлено зв'язки між ними.
4. **Проектування класів** - розроблено діаграму класів, яка відображає основні сутності системи та взаємозв'язки.

Проект "ViewTogether" демонструє ефективне застосування TypeScript для розробки сучасних веб-додатків. Використання строгої типізації забезпечує надійність коду та полегшує його підтримку. Архітектура додатку побудована з урахуванням сучасних практик розробки, таких як модульність, компонентний підхід та реактивне програмування.

Особливою цінністю запропонованого рішення є використання WebSocket для забезпечення комунікації в реальному часі без необхідності зберігання даних у базі даних, що робить додаток легким та швидким у розгортанні.