

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

Лабораторна робота № 4
з курсу «Розробка WEB-додатків (TypeScript)»
на тему:
«Розробка компонентів клієнтської взаємодії та візуалізації у веб-додатках
засобами TypeScript»

Виконав:

студент групи КН-41
Кадемський О. В.
«03» березня 2024 р.

Перевірів:

ст. викладач кафедри КНІС
к.т.н. Ізмайлов А. В.
«___» _____ 2024 р.

(оцінка, підпис)

м. Івано-Франківськ
2024

Мета роботи: Навчитись розробляти компоненти клієнтської взаємодії та візуалізації у веб-додатках засобами TypeScript.

1.1 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.

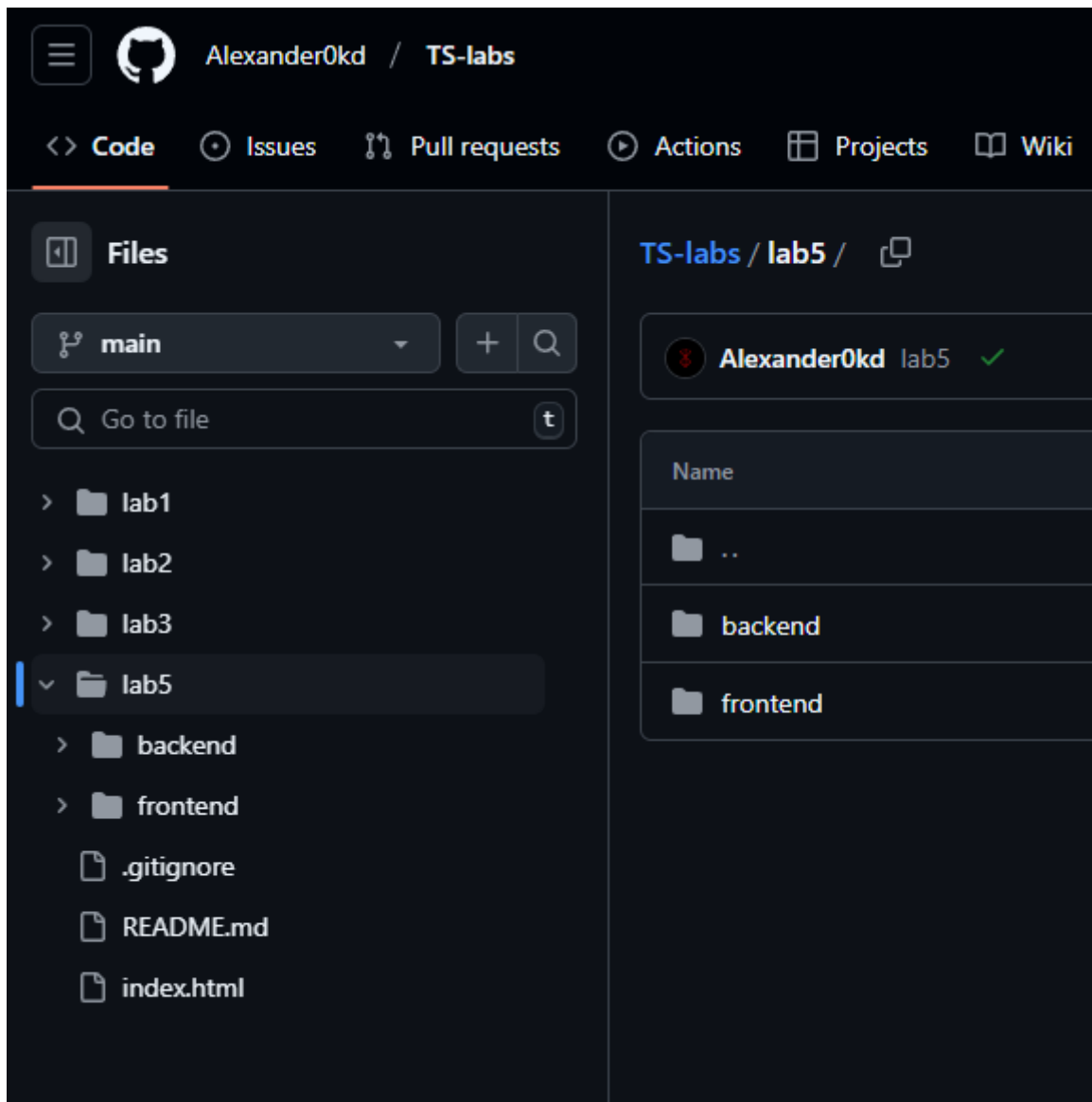
TS компілюється у JS і фактично є розширенням (superset) JS. Процес компіляції TS полегшує та стабілізує процес JS-розробки, оскільки дозволяє виявляти та виправляти помилки, які, у загальному випадку, довелося б виявляти уручну при традиційному процесі розробки у JS.

Класи у TS мають ряд корисних надбудов (у порівнянні з JS), які дозволяють ефективну реалізацію принципів ООП. Для обмеження доступу до властивостей об'єктів, у TS уведено наступні модифікатори видимості:

- `public` – властивість видима поза класом;
- `readonly` – значення властивості не може бути змінене;
- `private` – властивість видима лише у межах класу;
- `protected` – властивість видима лише у межах класу та класів-нащадків;

1.2 Хід роботи

1.2.1 У створеному на GitHub репозиторії для курсу створіть нову підпапку для завдання лабораторної:



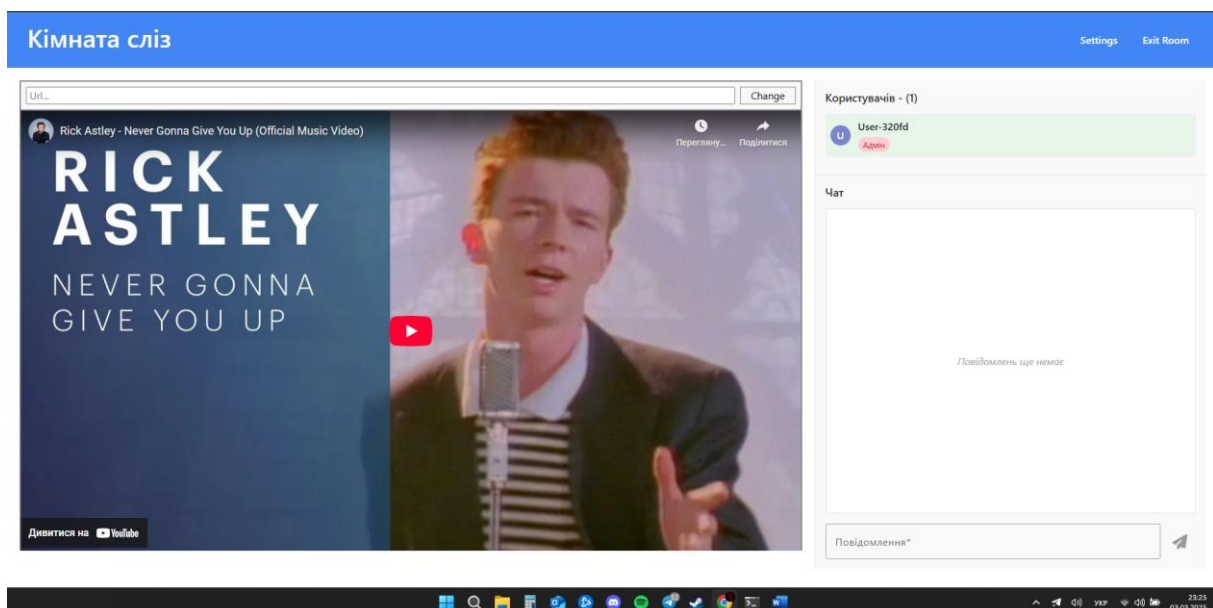
1.2.2 Для реалізації усіх сценарії (скриптів) слід використовувати TypeScript із явним вказанням типів, де це є можливо:

```
joinRoom(roomId: string): Observable<IRoom> {
    return this.webSocketService
        .createTwoSideWSConnection<
            IRoom,
            IJoinRoom
        >(WS_EVENT.CONNECTION.JOIN_ROOM, { roomId }, WS_EVENT.CONNECTION.ON_ROOM_JOINED, 'ONE')
        .pipe(
            switchMap((room) => {
                if (room) {
                    this.currentRoomUUID = room.uuid;
                    return this.subscribeToRoomUpdates();
                }

                return new Observable<IRoom>((observer) => {
                    this.currentRoomUUID = null;
                    observer.error(new Error('Room not found'));
                });
            })
        );
}

leaveRoom(roomId: string): void {
    this.currentRoomUUID = null;
    this.webSocketService.send<ILeaveRoom>(WS_EVENT.CONNECTION.LEAVE_ROOM, { roomId });
}
```

1.2.3 Реалізувати усі компоненти додатку, запропонованого у лабораторній роботі No2, які пов'язані із взаємодією із сервером (сервер може бути довільним, у т.ч. статичним) та маршрутизацією:



1.2.4 Надати викладачеві посилання на зроблену веб-сторінку, розміщену на сервері GitHub Pages (посилання повинне бути у форматі: <https://username.github.io/userrepo/userdirectory>), а також, посилання на безпосередньо сам репозиторій (посилання повинне бути у форматі: <https://github.com/username/userrepo>):

<https://github.com/Alexander0kd/TS-labs/tree/main/lab4>

<https://alexander0kd.github.io/TS-labs/lab4/frontend/dist/frontend/browser/>

Висновки: В ході виконання даної лабораторної роботи ми навчилися розробляти компоненти клієнтської взаємодії та візуалізації у веб-додатках засобами TypeScript.