

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЛЬВІВСЬКА ПОЛІТЕХНІКА**



**АВТОМАТИЗОВАНЕ  
ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ  
СИСТЕМ**

Лабораторна робота №1  
“Створення Github  
репозиторію”

**Виконав:**

студент гр. КІ-404  
Попов М. Ю.

**Прийняв:**

аспірант  
Федак П. Р.

## Завдання:

### 1. Створіть github-репозиторій з назвою

«csad<YY1YY2><група><прізвище студента  
ім'я><номер студента>», основна гілка develop.

YY1 - рік початку навчання

YY2 - рік закінчення навчання

Наприклад: 2019-2020 -> 1920

Повний приклад csad1920ki47spitzeras03

### 2. Додайте доступ для викладача.

### 3. Створіть гілку feature/develop/<номер завдання>.

Наприклад feature/develop/task1.

### 4. Створіть файл README з

a. деталями про репозиторій;

b. деталями завдання;

c. номер студента та дані з таблиці 1. Завдання студентів;

d. додайте інформацію про технологію, мову програмування та програмне забезпечення, які будуть використовуватися в наступних завданнях

будуть використані в наступних завданнях.

### 5. Створіть GIT TAG: <НАЗВА ПРОЕКТУ>\_<ВЕРСІЯ>\_WW<YYWWD>

· YY - поточний рік;

· WW - робочий тиждень;

· D - поточний день тижня.

### 6. Створіть pull-запит з іменем task1 і вкажіть викладача як рецензента.

### 7. Після того, як рецензент схвалив запит, перенесіть його у гілку розробки.

## Теорія:

**Репозиторій (Repository)** - це місце, де зберігаються всі файли проекту та історія їхніх змін. Репозиторій може бути локальним (на вашому комп'ютері) або віддаленим (на серверах, таких як GitHub, GitLab).

**Коміт (Commit)** - фіксація змін у файлах. Коміт містить знімок усіх файлів проекту в певний момент часу. Кожен коміт має унікальний хеш (ідентифікатор), за яким його можна ідентифікувати.

**Гілка (Branch)** - альтернативна версія історії комітів. Гілки дозволяють вести роботу над різними частинами проекту незалежно одна від одної. Основна гілка зазвичай називається main або master.

**Злиття (Merge)** - процес об'єднання двох гілок. Наприклад, коли завершена робота в гілці, її можна злити з основною гілкою.

**Форк (Fork)** - копія чужого репозиторію, що дозволяє вам експериментувати з кодом, не впливаючи на оригінальний проект. Зазвичай використовується для внесення змін до проектів з відкритим кодом.

**Клонування (Clone)** - створення локальної копії віддаленого репозиторію. За допомогою команди git clone можна завантажити весь проект із сервера на свій комп'ютер.

**Індекс (Staging Area)** - проміжна область, де фіксуються зміни перед створенням коміта. Файли додаються до індексу за допомогою команди git add.

**Теги (Tags)** - використовуються для позначення важливих точок в історії репозиторію, таких як випуски версій програмного забезпечення.

**Конфлікт (Conflict)** - виникає, коли Git не може автоматично об'єднати зміни в різних гілках через суперечливі зміни в одних і тих самих файлах. Потрібне ручне вирішення конфлікту.

**Віддалений репозиторій (Remote Repository)** - репозиторій, який знаходиться на сервері (GitHub, GitLab) і доступний для багатьох користувачів. Підключення до нього дозволяє спільну роботу над проектом.

**Пулл (Pull)** - операція, що об'єднує отримання змін з віддаленого репозиторію (git fetch) та злиття цих змін з поточною гілкою (git merge).

**Пуш (Push)** - відправлення локальних змін до віддаленого репозиторію. Це дозволяє оновити загальний репозиторій змінами, зробленими локально.

## Індивідуальне завдання:

Student number	Game	config format
16	rock paper scissors	XML

## Виконання

### 1) Створюю гітхаб репозиторій, та називаю його згідно вимог:

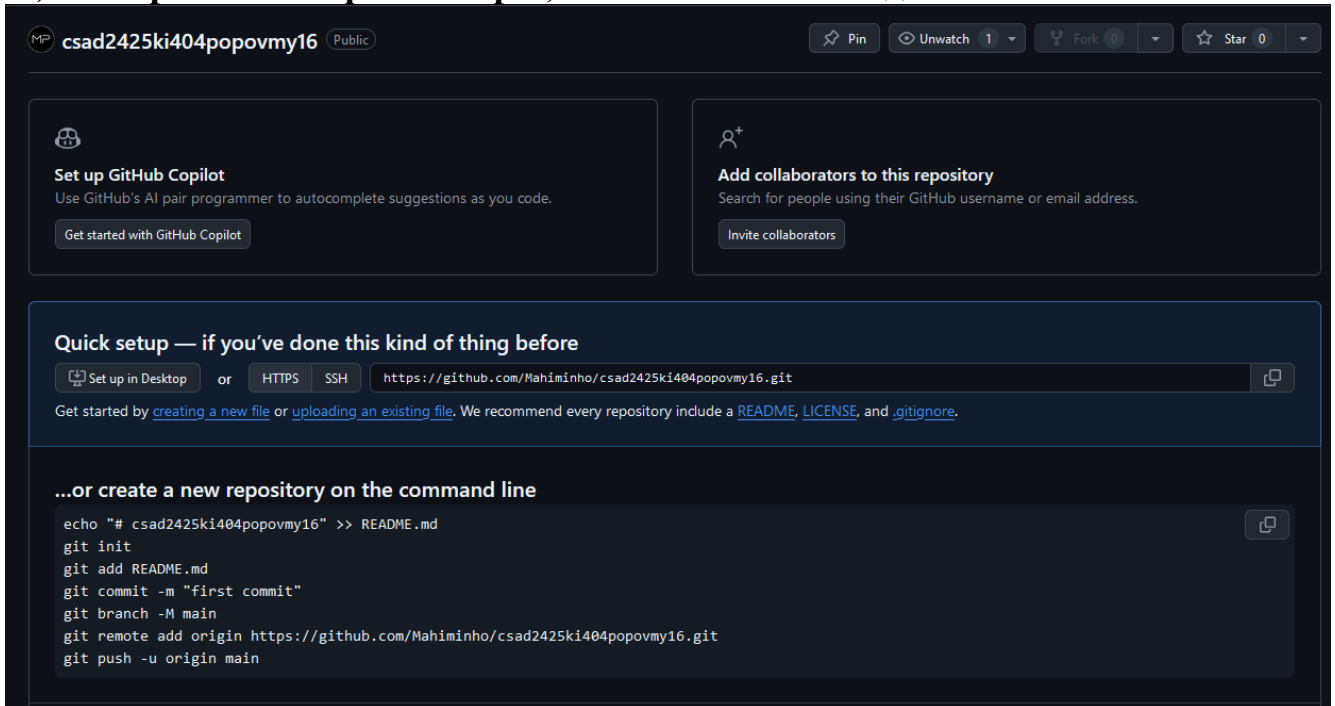


Рис. 1. Створення репозиторію та його іменування

### 2) Запрошую викладача в якості рев'юера в репозиторій:

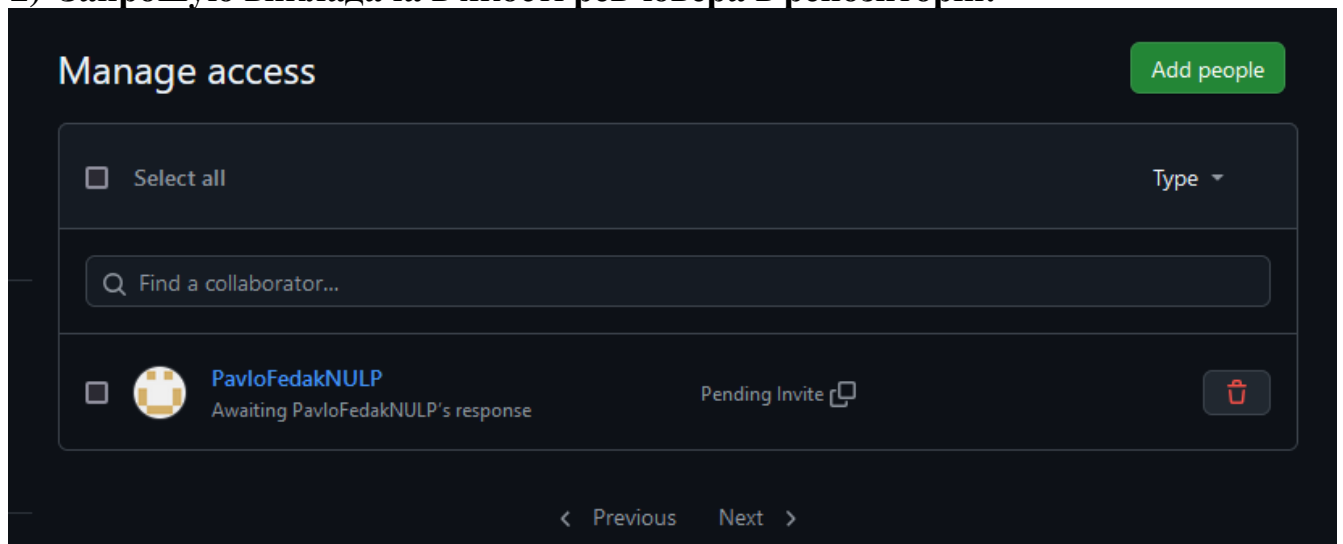


Рис. 2. Надання доступу викладачеві до репозиторію

### 3) Створюю гілку для першої лабораторної згідно вимог:

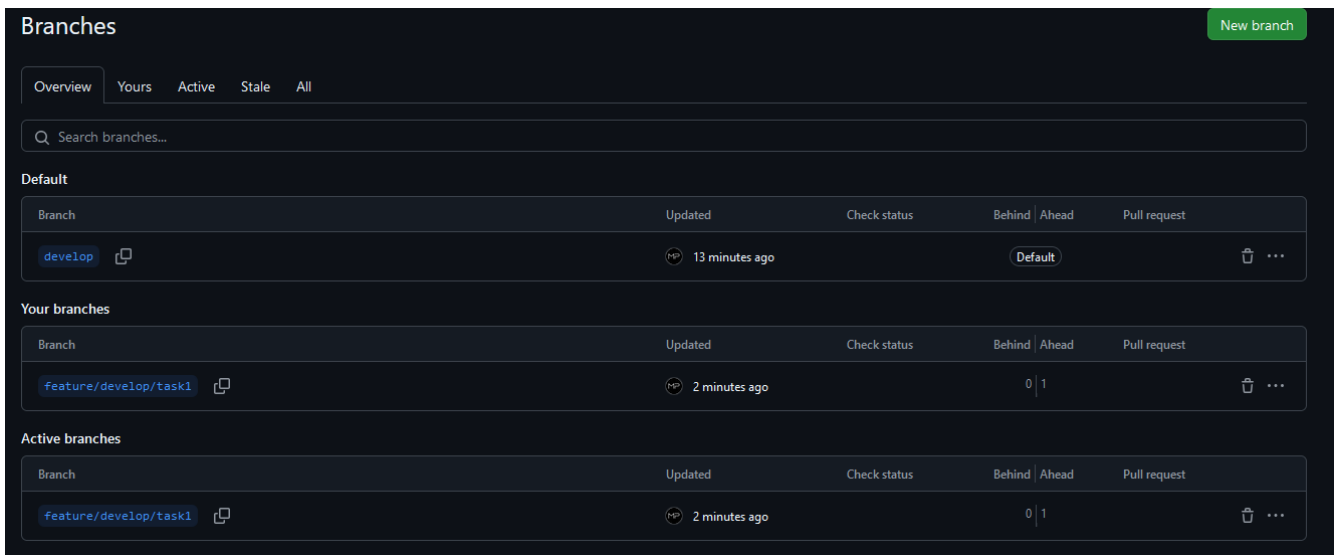


Рис. 3. Створення гілки feature/develop/task1

### 4) Створюю README.md файл з описом проекту:

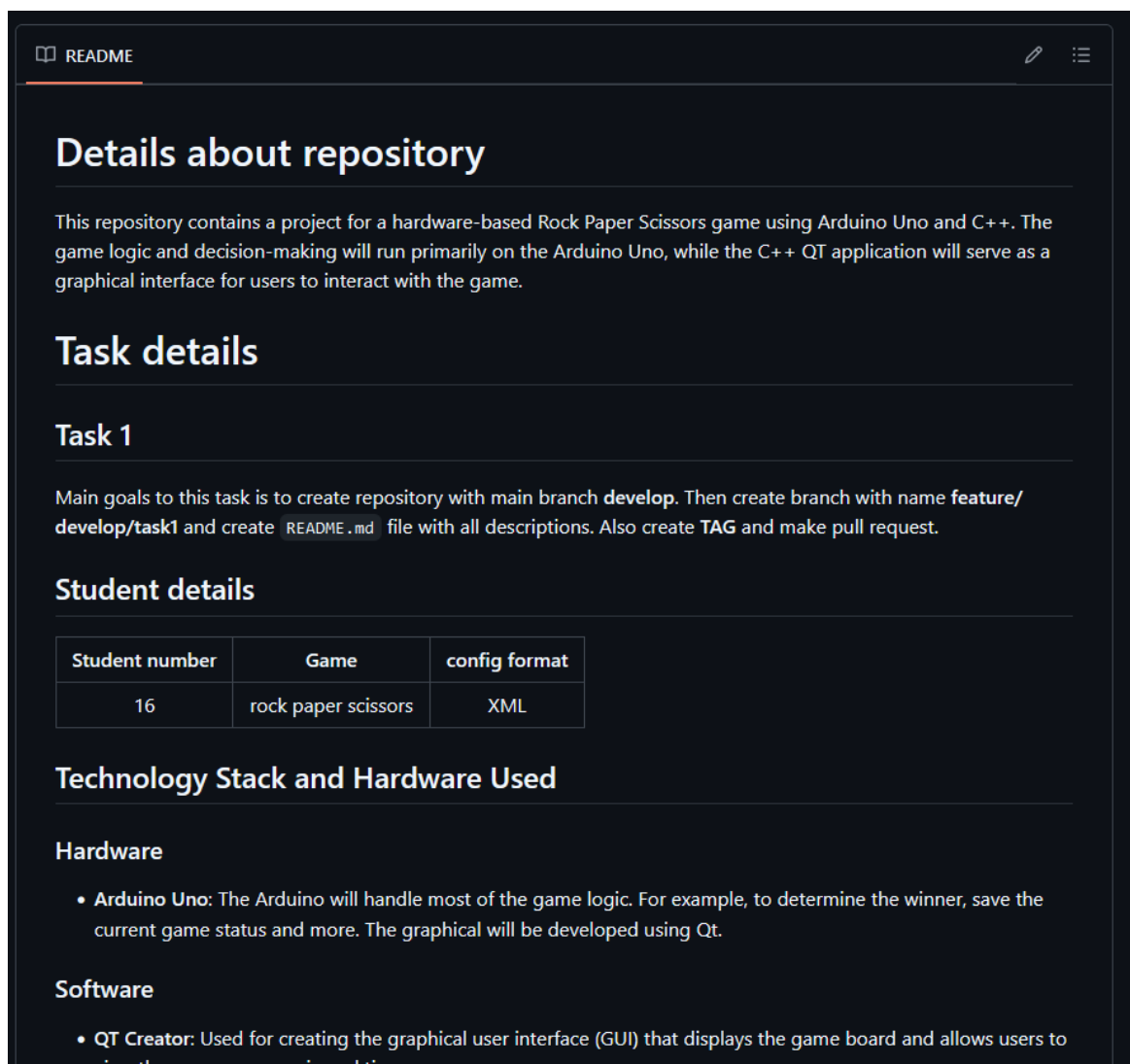


Рис. 4. Файл README.md з описом проекту

## 5) Створюю git tag згідно вимог:

```
MINGW64:/c:/Users/Sager/Desktop/CSAD_repo/csad2425ki404popovmy16
Sager@DESKTOP-2SCOFCT MINGW64 ~/Desktop/CSAD_repo/csad2425ki404popovmy16 (develop)
$ git tag RockPaperScissors_1.0_WW24026

Sager@DESKTOP-2SCOFCT MINGW64 ~/Desktop/CSAD_repo/csad2425ki404popovmy16 (develop)
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Mahiminho/csad2425ki404popovmy16
 * [new tag]          RockPaperScissors_1.0_WW24026 -> RockPaperScissors_1.0_WW24026

Sager@DESKTOP-2SCOFCT MINGW64 ~/Desktop/CSAD_repo/csad2425ki404popovmy16 (develop)
$ |
```

Рис. 5. Створення git tag: RockPaperScissors\_1.0\_WW24026

## 6) Створюю пул-реквест і отримую апрув від викладача:

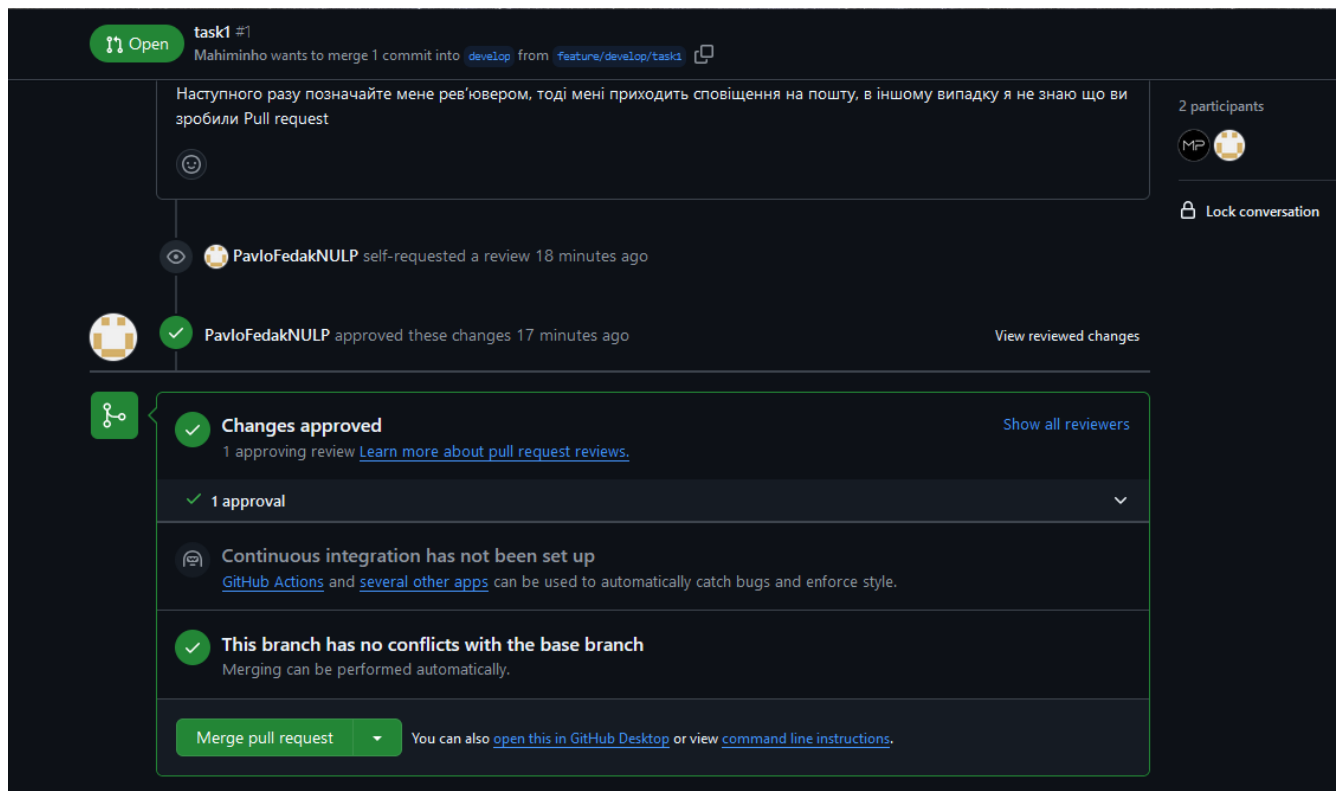
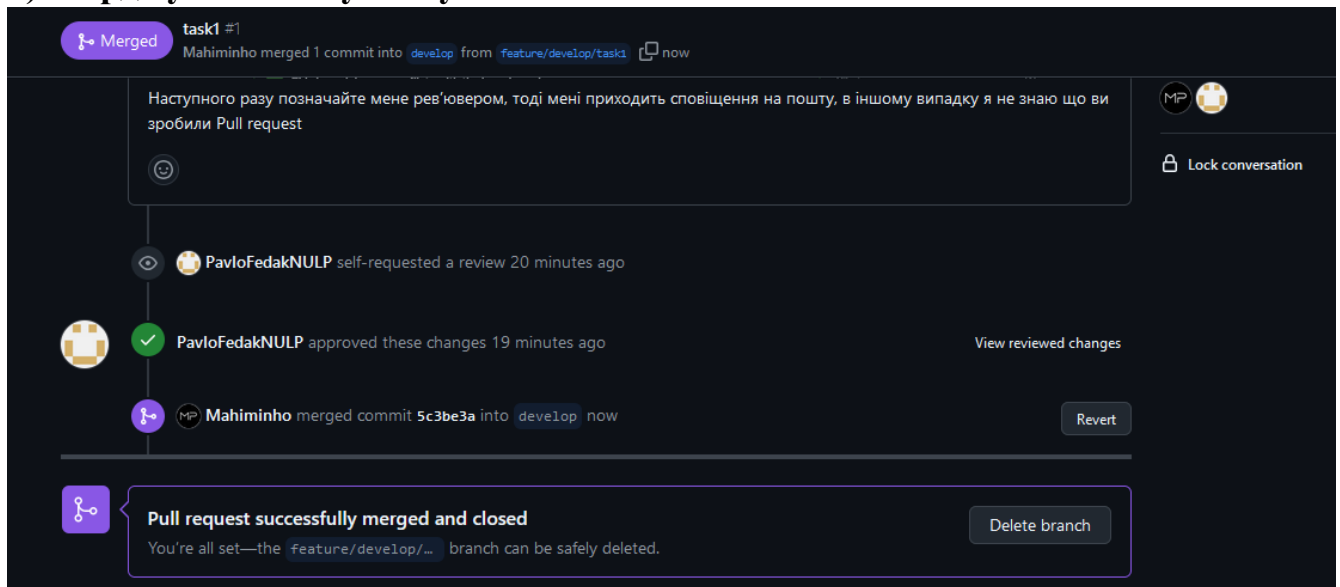


Рис. 6. Створення пул-реквесту та апрув від викладача

## 7) Мерджу в головну гілку:



*Рис. 7.* Перенесення змін з гілки feature/develop/task1 в develop

### Висновок:

Впід час виконання даної лабораторної роботи, я навчився створювати репозиторії, додавати до нього ревіювера, створювати нові гілки та теги, а також робити пул-реквести та мерджити.

### Список використаних джерел:

1. GitHub Docs: <https://docs.github.com/>
2. Instructions for practical tasks and coursework from “Computer systems automated design”