

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



**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ**  
**КОМП'ЮТЕРНИХ СИСТЕМ**

Task1 Initiate GIT repository

Виконав :  
Студент групи КІ-401  
Онисько М.М.  
Прийняв:  
Федак П.Р

**2024**

## **Завдання:**

1. Створити репозиторій github
2. Додати доступ для лектора.
3. Створити гілку feature/develop/<task number>.
4. Створити файл README за допомогою:
  - a. подробиці про репозиторій;
  - b. деталі завдання;
  - в. номер студента та реквізити з таблиці 1. Завдання студентів;
  - с. додати деталі про технологію, мову програми та апаратне забезпечення, яке буде використовувати в наступних завданнях
5. Створіть тег GIT
6. Створіть запит на отримання з назвою task1 і призначити лектора як рецензента.
7. Після схвалення рецензентом – об'єднати запит у гілку розробки.

## **Теоретичні відомості:**

### **1. Ініціалізація репозиторію GitHub**

Ініціалізація репозиторію – це створення нового сховища для збереження коду та контролю його версій. GitHub – це популярна платформа для хостингу Git-репозиторіїв, яка дозволяє розробникам співпрацювати над проєктами, відстежувати зміни та управляти версіями.

## **Основні кроки:**

- **Створення нового репозиторію на GitHub:**
  - Зайдіть у свій акаунт на GitHub і натисніть кнопку "New Repository".
  - Дайте назву репозиторію, за бажанням додайте опис та виберіть, чи буде репозиторій публічним або приватним.
  - Опціонально можна додати .gitignore файл для виключення певних файлів з репозиторію, та README.md файл, що міститиме інформацію про проєкт.
- **Ініціалізація локального репозиторію:**
  - Створіть нову директорію для проєкту та ініціалізуйте в ній Git:  
  
`git init`
  - Додайте віддалений репозиторій на GitHub:  
  
`git remote add origin https://github.com/username/repository.git`

- Додайте файли до індексу та виконайте перший коміт:

```
git add .  
git commit -m "Initial commit"
```

- Відправте зміни на GitHub:

```
git push -u origin main
```

## 2. Гілки (Branches)

Гілки дозволяють розробникам працювати над різними частинами проєкту паралельно, не впливаючи на основний код до моменту злиття.

### Основні команди:

- **Створення нової гілки:**

```
git checkout -b new-branch
```

Ця команда створює нову гілку та перемикає на неї.

- **Перемикання між гілками:**

```
git checkout branch-name
```

- **Перегляд всіх гілок:**

```
git branch
```

- **Злиття гілки (Merge):** Для злиття гілки new-branch в основну (наприклад, main):

```
git checkout main  
git merge new-branch
```

## 3. Pull Request (Запит на злиття)

Pull Request – це механізм GitHub для внесення змін у код. Використовується, коли одна гілка готова до злиття з іншою, зазвичай після завершення розробки нової функції або виправлення помилки.

### Основні етапи:

1. **Внесення змін у гілку:** Розробник створює окрему гілку для роботи над зміною.
2. **Пуш змін у віддалений репозиторій:** Зміни завантажуються на GitHub.

```
git push origin new-branch
```

### 3. Створення Pull Request:

- Перейдіть на GitHub, знайдіть вашу гілку та натисніть "Compare & pull request".
  - Додайте опис, чому ці зміни потрібні.
4. **Огляд та обговорення:** Інші учасники проекту можуть переглянути зміни, додати коментарі та залишити відгуки.
5. **Злиття PR:** Після затвердження змін Pull Request можна злити в основну гілку.

### **Виконання:**

1. Створив Git репозиторій з назвою «csad2425ki401onysko15»
2. Додав лектора як collaborator
3. Скачав репозиторій на комп'ютер, та створив гілку feature/develop/task1 від default гілки develop
4. Створив README.md файл згідно вимог.

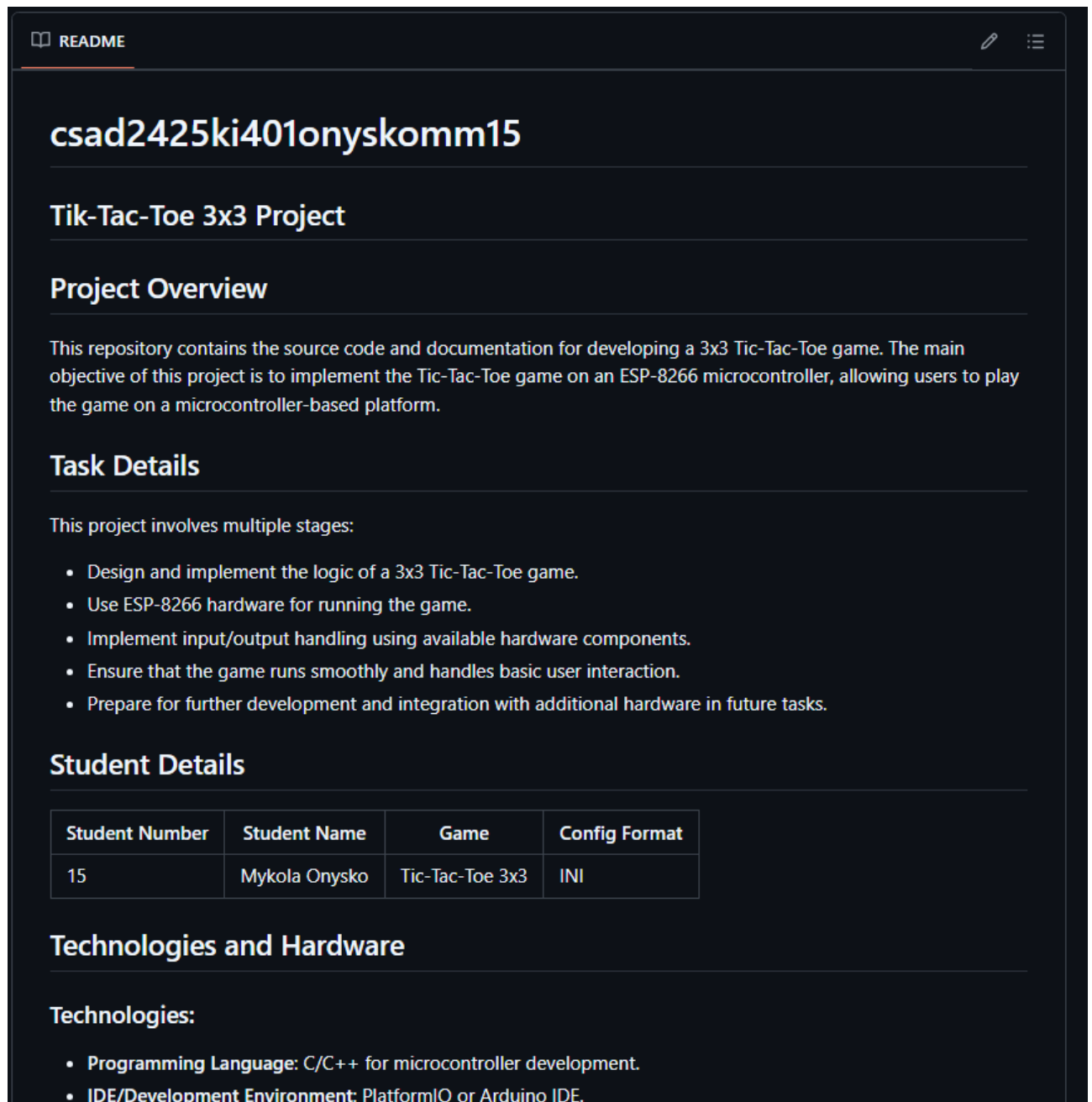


Рис.1 – Вміст README файлу

- Зауважив зміни в репозиторій та створив Pull Request з назвою task1 та лектором як рецензентом
- Об'єднав створену гілку з гілкою develop

### Висновки:

На даній лабораторній роботі я ініціалізував GitHub репозиторій. Створив гілку з завданням, де описав майбутній проект, та зробив Pull Request з цією гілкою.

### Посилання:

- GitHub репозиторій- <https://github.com/onskmk1/csad2425ki401onyskomm15>

2. **GitHub Docs** (2023). GitHub Documentation: Working with GitHub - <https://docs.github.com/>
3. **Dabbagh, N., & Kitsantas, A.** (2012). Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *The Internet and Higher Education*, 15(1), 3-8.
4. **Loeliger, J., & McCullough, M.** (2012). *Version Control with Git* (2nd ed.). O'Reilly Media.
5. **Chacon, S., & Straub, B.** (2014). *Pro Git* (2nd ed.). Apress.