**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЛЬВІВСЬКА ПОЛІТЕХНІКА**



**АВТОМАТИЗОВАНЕ**

**ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ**

**СИСТЕМ**

Лабораторна робота №3

“Створення клієнт-серверної гри Камінь-Ножиці-Папір”

**Виконав:**

студент гр. KI-401

Гавриляк Д. В.

**Прийняв:**

Федак П. Р.

Львів – 2024

**Завдання:**

1. Develop Server and Client.

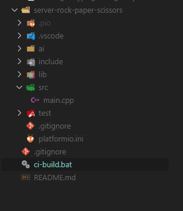
**Індивідуальне завдання:**

Згідно списку групи я маю 14 варіант, так як я маю такий порядковий номер у списку.



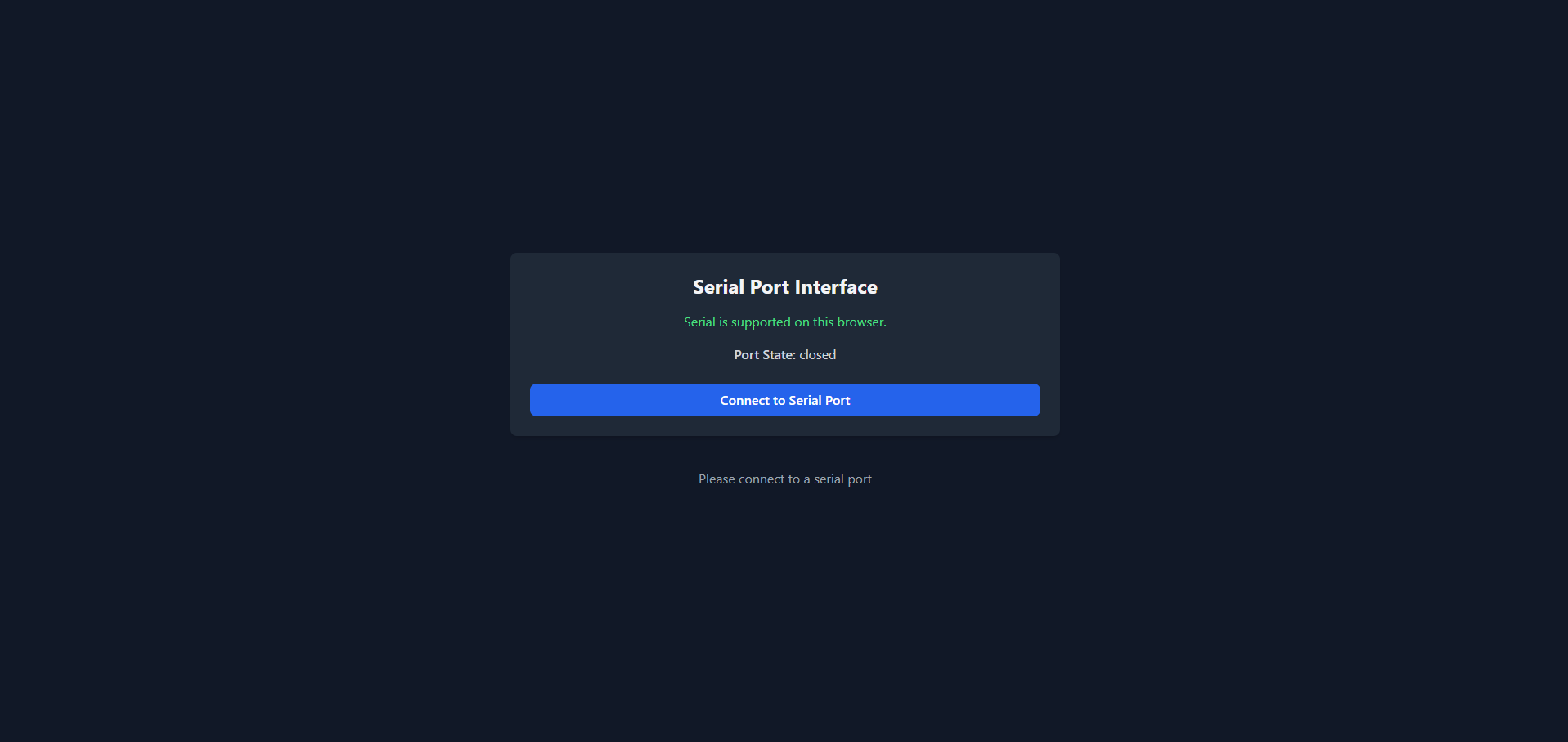
**Виконання:**

На даному скріншоті продемонстровано оновлена структура проекту, були добавлені нові компоненти для клієнта і нові файли для сервера, а також файл ci-build.bat

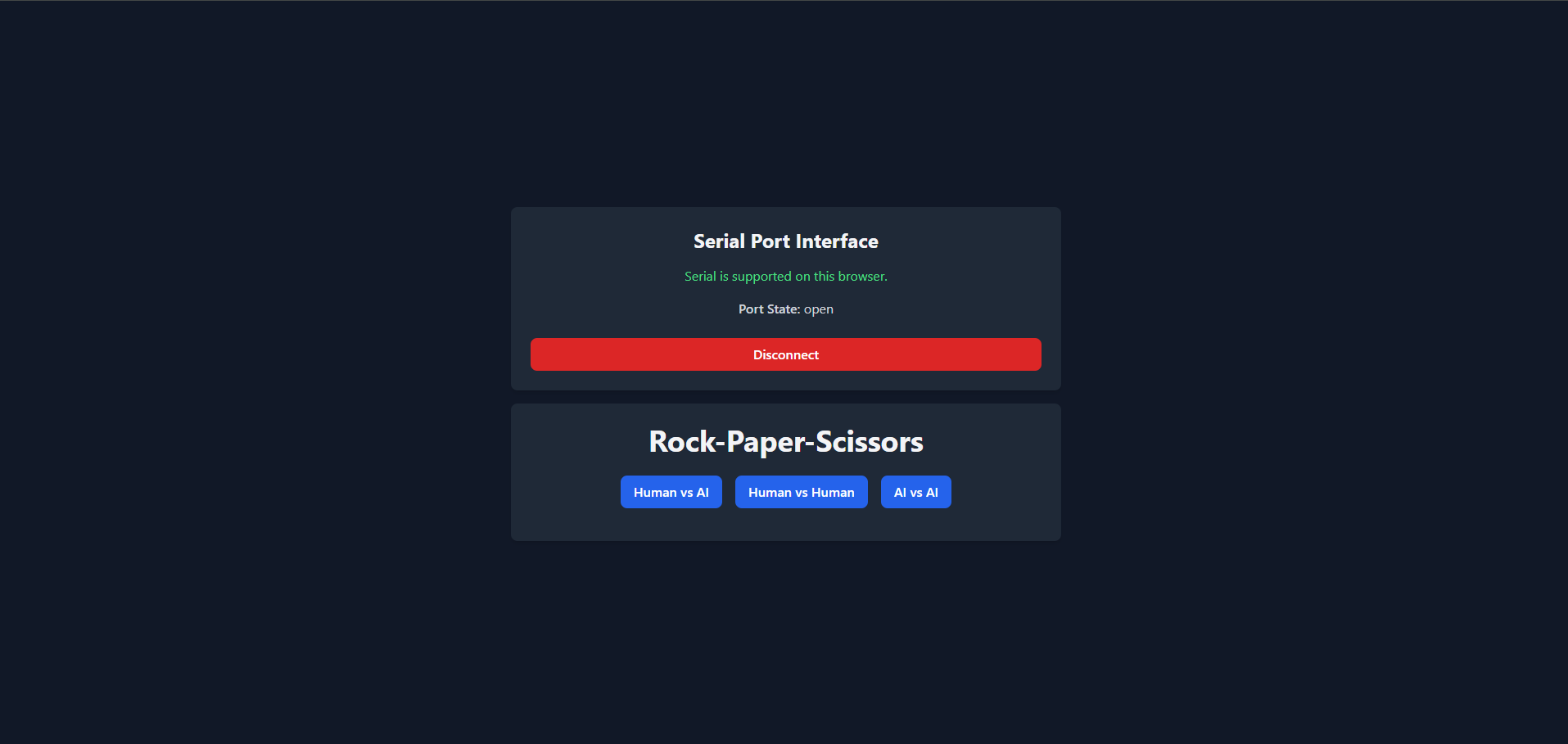


***Рис.2.*** ci-build.bat

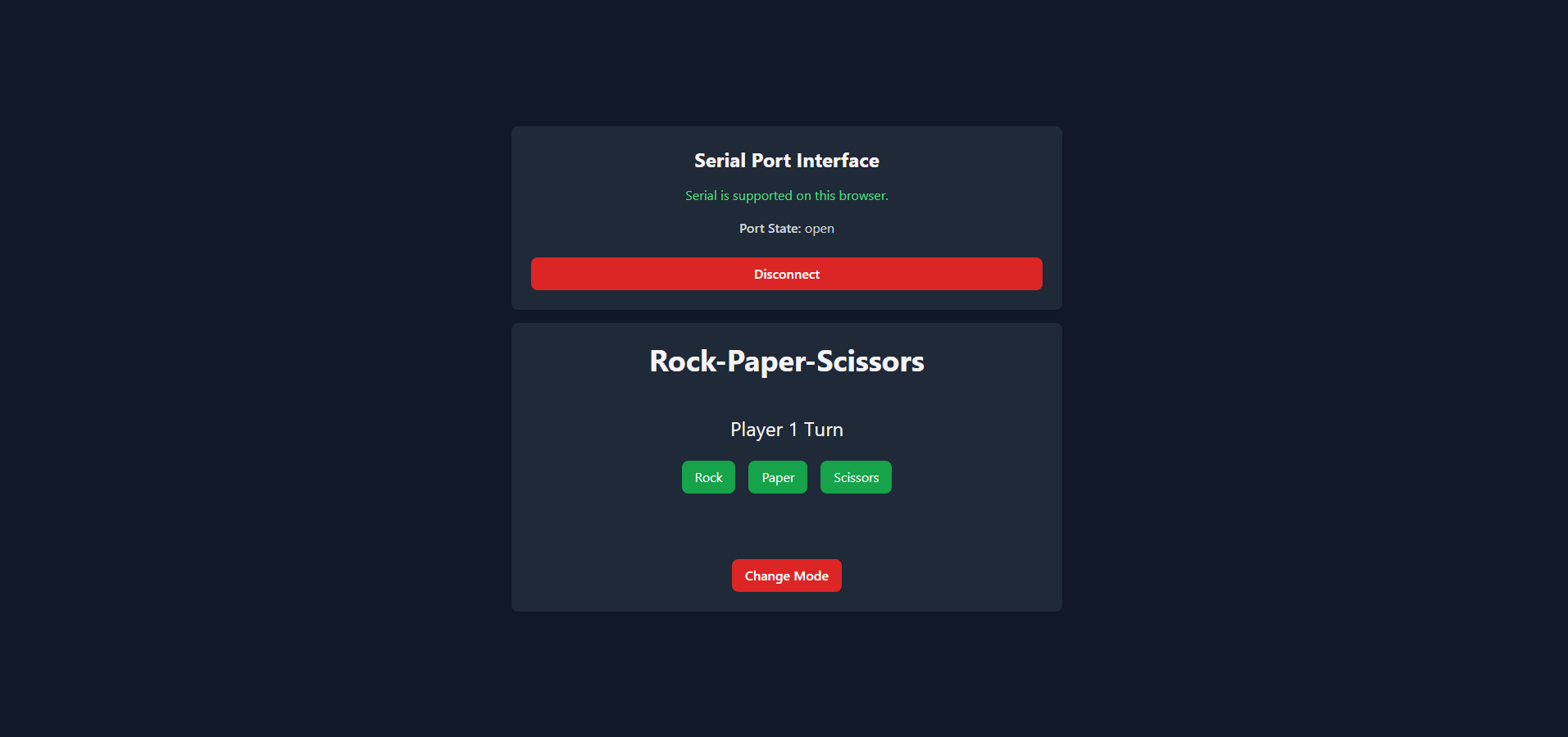
Файл з написаним flow для роботи тестів і компіляції проектів з подальшим виведенням скомпільованих файлів у артефакти



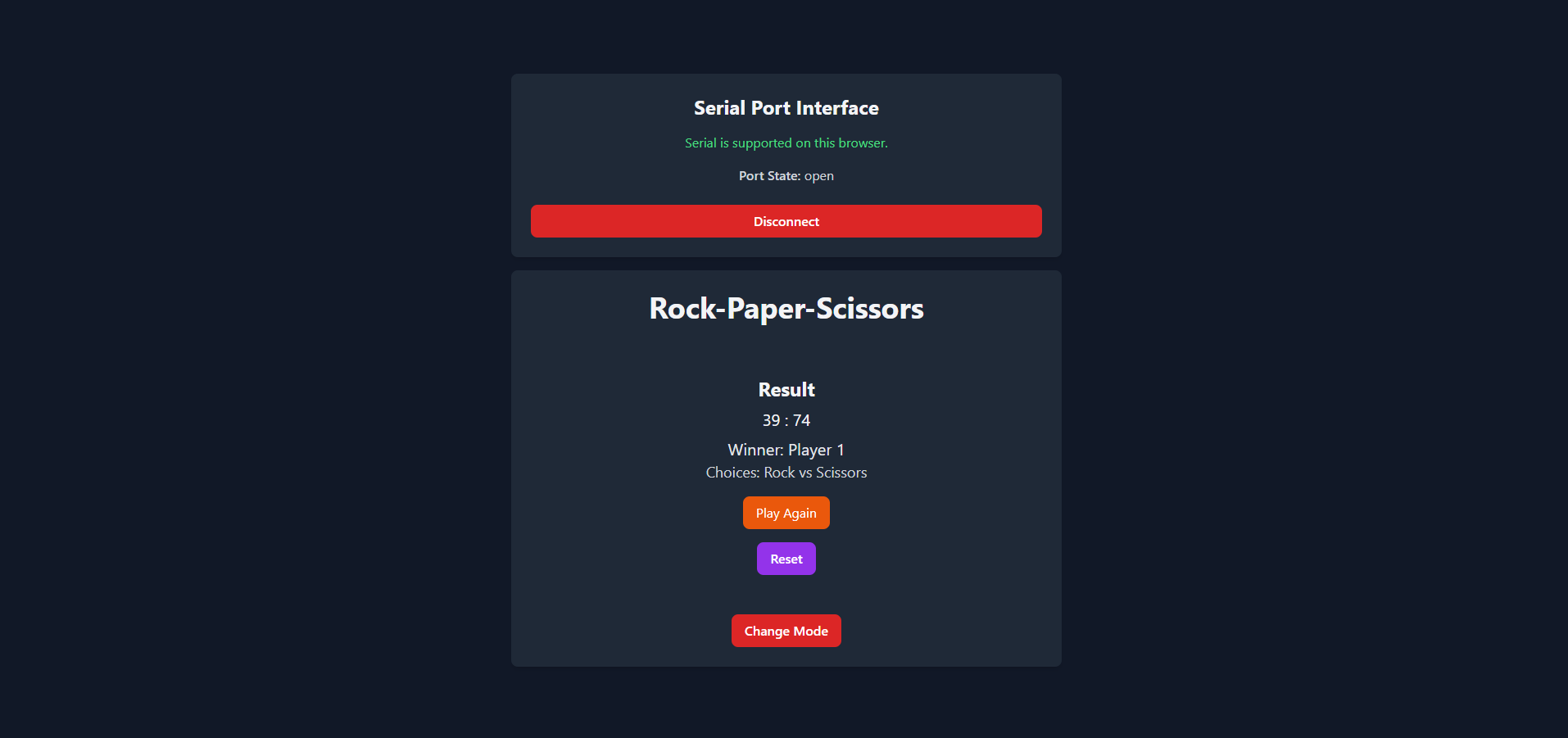
***Рис.3.*** Ui клієнту, підключення плати



***Рис.4.*** Ui клієнту, вибір режиму



***Рис.5.*** Ui клієнту, хід гравця



***Рис.6.*** Ui клієнту, результат гри

Код сервера (плати Arduino Nano) з нейронною мережею:

int lastMoves[13];

// Масиви ваг для нейронної мережі

float inputWeights[3][8] = {

// Значення ваг для вхідного шару

};

float hiddenWeights[8][3] = {

// Значення ваг для прихованого шару

};

float hiddenBias[8] = {

// Зміщення для нейронів прихованого шару

};

float outputBias[3] = {

// Зміщення для нейронів вихідного шару

};

// Функція для перетворення числа на назву ходу (камінь, папір, ножиці)

String moveToString(int choice) {

if (choice == 0) {

return "Rock"; // Якщо choice дорівнює 0, повертаємо "Rock"

} else if (choice == 1) {

return "Paper"; // Якщо choice дорівнює 1, повертаємо "Paper"

} else {

return "Scissors"; // В іншому випадку повертаємо "Scissors"

}

}

// Функція активації ReLU (Rectified Linear Unit)

float relu(float x) {

return x > 0 ? x : 0; // Повертає максимальне значення між x та 0

}

// Функція softmax для нормалізації вихідних значень

void softmax(float x[], int size) {

// Знаходимо максимальне значення в масиві x

float max\_val = x[0];

for (int i = 1; i < size; i++) {

if (x[i] > max\_val) {

max\_val = x[i];

}

}

// Обчислюємо суму експонент

float sum = 0;

for (int i = 0; i < size; i++) {

sum += exp(x[i] - max\_val);

}

// Нормалізуємо значення

for (int i = 0; i < size; i++) {

x[i] = exp(x[i] - max\_val) / sum;

}

}

**Створив pull-запит з іменем Task 3 with smart ai і дочекався апруву від викладача**

**Після того, як рев’ювер схвалив запит, переніс його у гілку розробки**

**Висновок:**

Під час виконання даної лабораторної роботи, я дізнався як писати локальний СІ скрипт для локальної збірки проекту та запуску тестів. Також написав гру камінь-ножиці-папір, яка має 3 режими: людина проти АІ, людина проти людини та АІ проти АІ. Для АІ, який використовується в першому режимі я натренував нейронну мережу та реалізував її на сервері.