

Міністерство освіти і науки України  
Київський фаховий коледж туризму та готельного господарства  
Відділення підприємництва та інформаційних технологій  
Циклова комісія інформаційних технологій

**Звіт**  
з навчальної практики  
Вступ до фаху

**Виконав:**  
студент КІ-24 групи  
Лівіцький Кирило  
**Перевірили керівники практики:**  
Любима А.Є., Панібратов А.І.

Київ 2025 рік

## ВСТУП

Практична підготовка студентів є складовою частиною освітнього процесу і спрямована на оволодіння студентами системою професійних вмінь і навичок, а також первинним досвідом професійної діяльності, і має сприяти саморозвитку студента. Практична підготовка покликана не тільки забезпечити формування професійних вмінь, але й професійних навичок.

Метою проведення навчальної практики «Вступ до фаху» для студентів спеціальності «Комп'ютерна інженерія» є розкриття змісту роботи майбутнього фахівця з комп'ютерної інженерії; формування професійного світогляду майбутнього фахівця з комп'ютерних технологій у бізнесі, цілісне уявлення про його сутність та роль в сучасному суспільстві, формування цілісного представлення про суть надання послуг в цифровій та апартній сфері в цілому: актуальність проектування комп'ютерних мереж сучасними засобами зв'язку; діагностування несправностей роботи комп'ютерних систем.

## ЗАВДАННЯ

**Завдання 1.** Провести smoke, функціональне, UX-тестування, тестування безпеки, стрес-тестування предмета

**Предмет тестування:** олівець

### **Smoke тести**

Для smoke-перевірки обрано наступний кейс.

**Перевірка здатності олівця писати:**

1. Беремо олівець.
2. Перевіряємо, чи він заточений.
3. Малюємо лінію на аркуші паперу.
4. Перевіряємо, чи лінія є видимою.

**Очікувана поведінка:** олівець залишає чітку і рівномірну лінію.

### **Позитивні тести**

**Перевірка якості письма олівця:**

1. Беремо аркуш паперу.
2. Малюємо лінії різного натиску (легкий, середній, сильний).
3. Перевіряємо товщину і якість ліній.

**Очікувана поведінка:** олівець пише рівномірно, лінії різної товщини відповідають рівню натиску.

**Перевірка роботи гумки (якщо є):**

1. Написати кілька слів олівцем.
2. Використати гумку на протилежному кінці олівця.
3. Перевірити, чи текст стерся без залишків слідів.

**Очікувана поведінка:** текст зникає, папір не пошкоджений.

**Перевірка міцності корпусу:**

1. Беремо олівець і стискаємо його руками.
2. Наносимо легкі удари олівцем об стіл.

**Очікувана поведінка:** корпус олівця не тріскається і не деформується.

### **Негативні тести**

**Спроба писати тупим олівцем:**

1. Використовуємо олівець без заточення.
2. Малюємо лінії на папері.

**Очікувана поведінка:** олівець не залишає чітких слідів, лінії стають нерівними.

**Ломка грифеля під час письма:**

1. Натискаємо на олівець із надмірною силою під час письма.

**Очікувана поведінка:** грифель ламається.

**Використання олівця в екстремальних умовах:**

1. Пишемо на вологому папері.
2. Використовуємо олівець на нерівній поверхні (наприклад, дерев'яній дошці).

**Очікувана поведінка:** якість письма значно погіршується або олівець не пише.

**Навантажувальні тести**

**Багаторазове письмо:**

1. Малюємо лінії безперервно протягом 30 хвилин.
2. Оцінюємо стан грифеля і зношеність гумки (якщо є).

**Очікувана поведінка:** олівець поступово стирається, але не ламається.

**Кількаразове заточування:**

1. Заточуємо олівець до повного зникнення грифеля.
2. Вимірюємо, скільки заточувань витримує корпус.

**Очікувана поведінка:** корпус не розвалюється до закінчення грифеля.

**UX-тестування**

1. Перевіряємо, чи зручно тримати олівець у руці.
2. Оцінюємо, чи ковзає олівець під час письма.
3. Перевіряємо, чи зручно заточувати олівець стандартною точилкою.
4. Оцінюємо, чи зручно стирати написане гумкою.

**Очікувана поведінка:**

- Олівець зручно тримати і використовувати.
- Грифель не кришиться, гумка ефективно стирає текст.

**Тестування безпеки**

1. Перевіряємо, чи корпус олівця не містить гострих країв, які можуть порізати руку.
2. Перевіряємо, чи не токсичні матеріали, з яких виготовлений олівець (наприклад, лак на корпусі).
3. Оцінюємо, чи олівець безпечний для дітей (немає дрібних деталей, які можна проковтнути).

**Очікувана поведінка:** олівець відповідає всім нормам безпеки.

**Висновок:** Олівець пройшов перевірки на здатність писати, зручність використання, міцність та безпеку. Він показав хорошу якість письма, витримав стандартні навантаження та виявився зручним у використанні. Невеликі недоліки, як ламкість грифеля при сильному натиску, є типовими для такого типу інструментів. Загалом, олівець відповідає очікуванням.

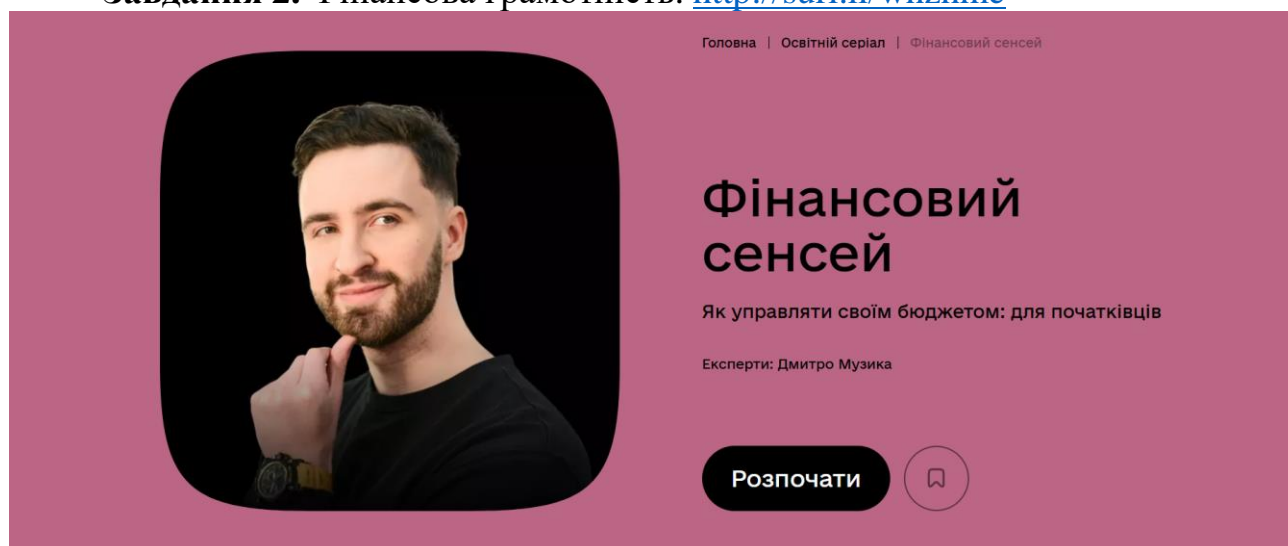
Завдання 2. Фінансова грамотність. <http://surl.li/wnzhme>

Рис. 1. Головний екран тесту “Фінансовий сенсей”



## Вітаємо! Ви успішно пройшли тест «Фінансовий сенсей»

Вручаємо сертифікат, що підтверджує ваші знання та навички!  
Можете завантажити його, поділитись із друзями в соцмережах, також сертифікат доступний у розділі «Моя Освіта»

Повернутись до серіалу

Завантажити сертифікат

Рис. 2. Успішне проходження тесту “Фінансовий сенсей”

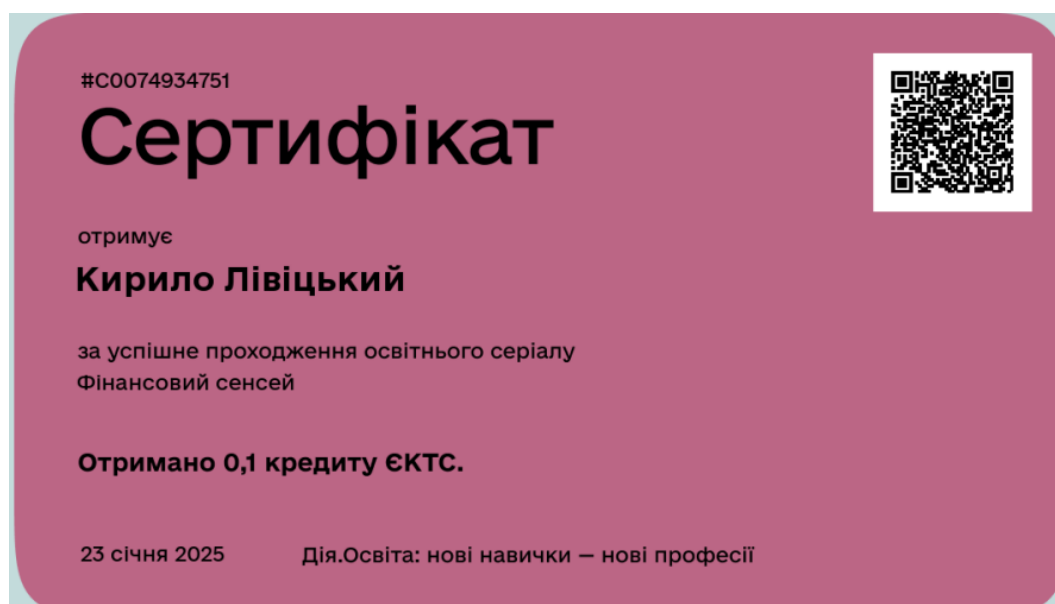


Рис. 3. Сертифікат за успішне проходження тесту “Фінансовий сенсей”

**Висновок:** Було пройдено тест на фінансову грамотність. Тепер я знаю більше про фінанси, як ними користуватися і все, що з ними пов'язано.

**Завдання 3.** Завантажте GitHub Desktop. Зареєструйтеся на сервісі і встановіть застосунок собі на ПК. Створіть свій власний репозиторій та завантажте до нього папку з певними файлами. Змініть складові папки та завантажте оновлену версію у репозиторій.

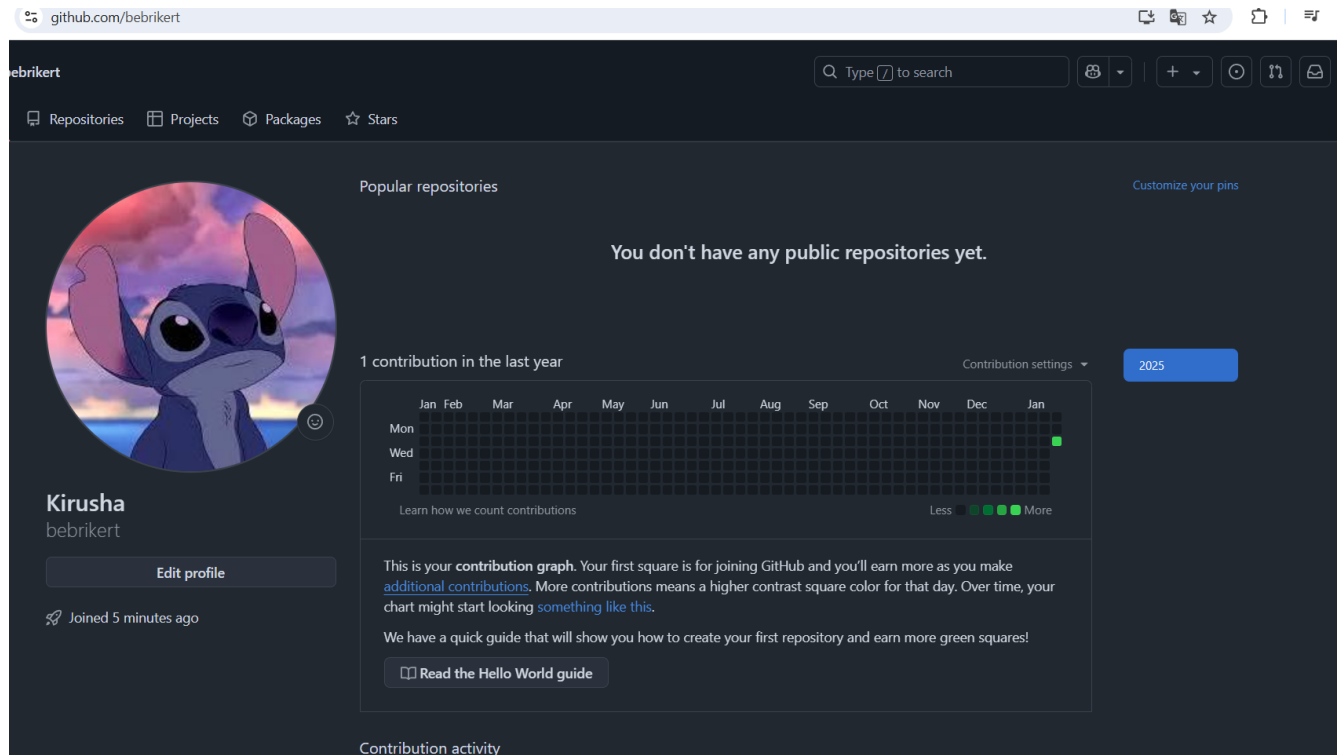


Рис.4. Сторінка профіля - <http://surl.li/ztrgal>

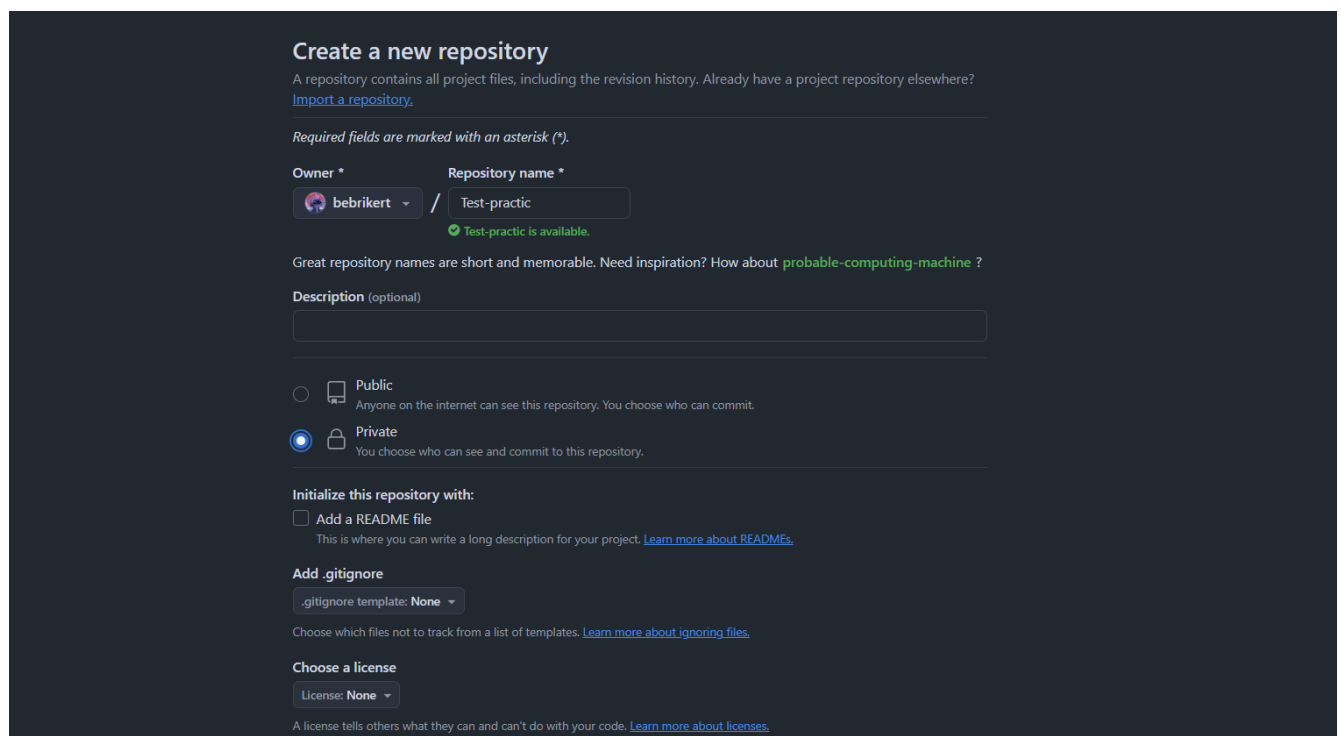


Рис. 5. Процес створення репозиторію

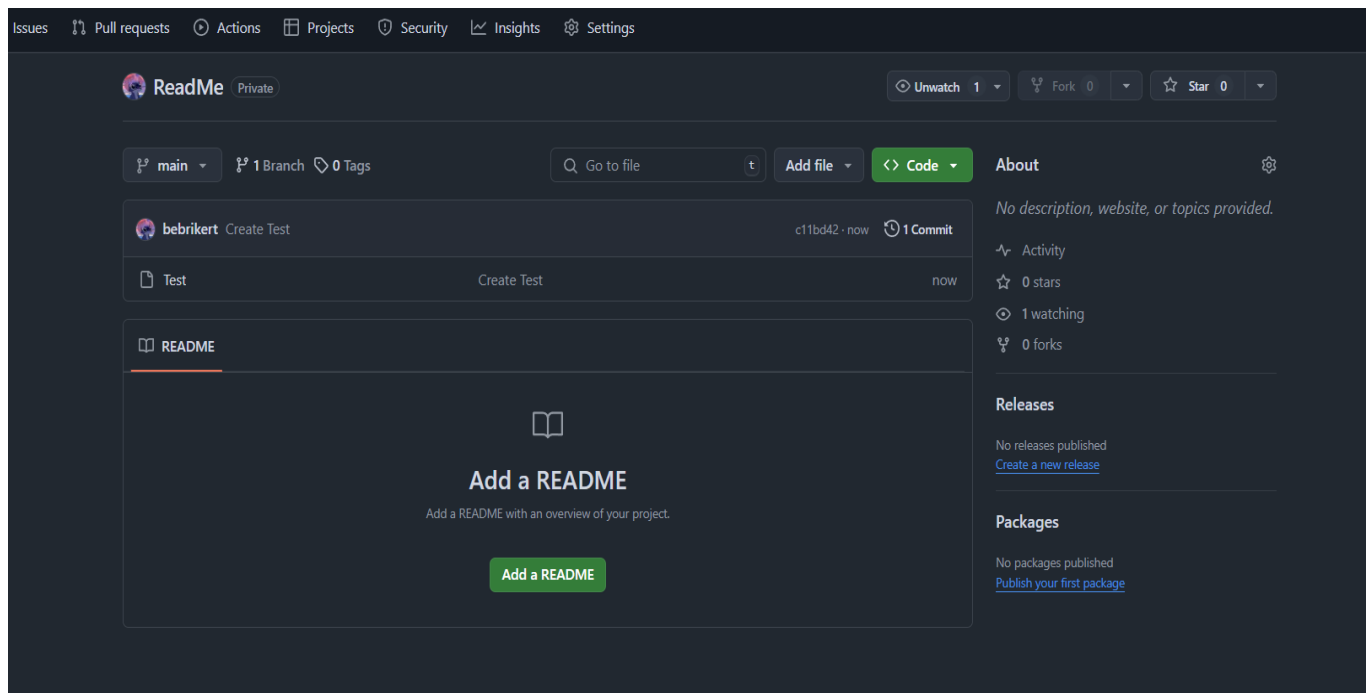


Рис. 6. Створений репозиторій - <http://surl.li/axqovj>

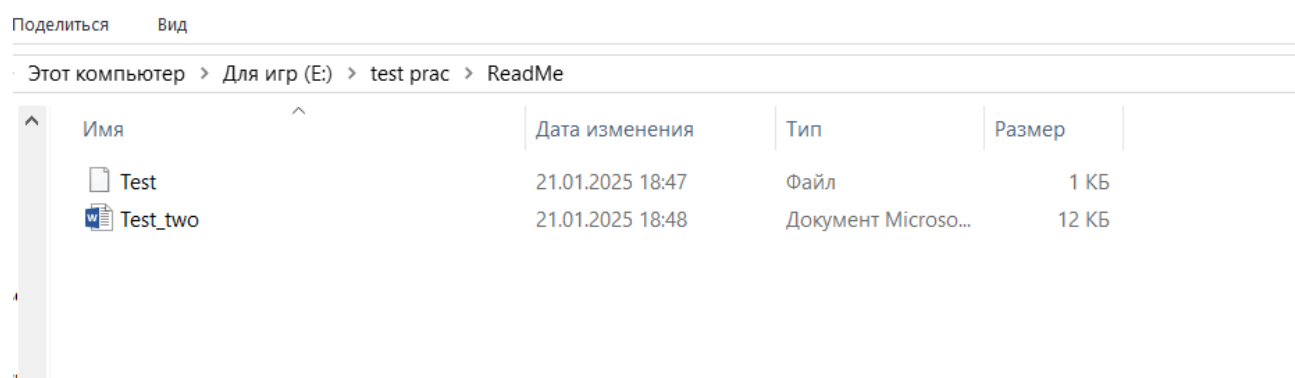


Рис.7. Додавання нового файлу в репозиторій

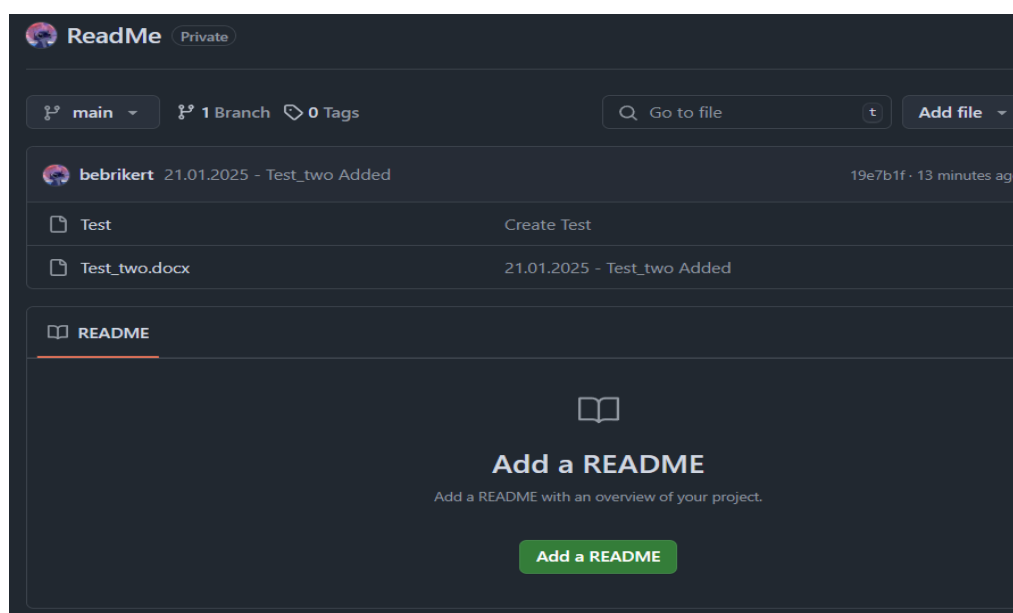


Рис. 8. Оновлена версія репозиторія

**Висновок:** Я дізнався, як користуватися GitHub створювати в ньому репозиторії в яких можна тримати різні файли для своїх потреб, а також вона може бути, як сторінка твоїх робіт і досягнень, які можуть бути корисним при вступленні в різні компанії.

#### Завдання 4.

Вивчіть поняття тестової документації. Завантажте тестовий додаток, та спробуйте знайти якомога більше помилок у ньому. Розмістіть створену тестову документацію на своєму GitHub репозиторії.

##### Поняття тестової документації:

**Test-Case** - це покроковий опис дій, які необхідно виконати для тестування програми та її функцій.

**Test Plan** - Це документ, в якому описується весь обсяг тестових робіт, починаючи від опису об'єкта, стратегії, графіка, критеріїв початку і закінчення випробувань, закінчуючи необхідним в процесі експлуатації обладнанням, спеціальними знаннями, а також оцінкою ризиків з варіантами їх вирішення.

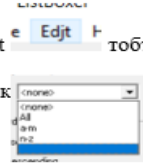
**Bug Report** -технічний документ, звіт про помилки (дефекти) у роботі об'єкту тестування (застосунку, фічі, сайту), що складає тестувальник. У ньому міститься опис ситуації чи послідовність дій, яка призвела до виникнення дефекту ПЗ, а також зазначені причини та очікуваний результат.

**Check List** - це документ із прописаними діями, які слід по черзі виконати.

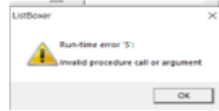
ASYCFILT.DLL	06.10.1998 01:00	Расширение при...	145 КБ
COMCAT.DLL	31.05.1998 01:00	Расширение при...	22 КБ
COMDLG32.OCX	24.06.1998 01:00	Элемент ActiveX	137 КБ
ListBoxer	10.09.2024 11:27	Архив WinRAR	1 442 КБ
Listboxer.cnt	12.05.1999 17:58	Файл "CNT"	1 КБ
ListBoxer	12.05.1999 18:22	Приложение	60 КБ
LISTBOXER	12.05.1999 18:09	Файл справки	13 КБ
MSVBVM60.DLL	25.06.1998 01:00	Расширение при...	1 376 КБ
OLEAUT32.DLL	06.10.1998 01:00	Расширение при...	585 КБ
OLEPRO32.DLL	06.10.1998 01:00	Расширение при...	161 КБ
Readme	04.05.1999 13:28	Текстовый докум...	1 КБ
setup	10.09.2024 11:27	Приложение	138 КБ
SETUP.LST	10.09.2024 11:27	Файл "LST"	4 КБ
SETUP1	20.06.1998 01:00	Приложение	280 КБ
ST6UNST	18.06.1998 01:00	Приложение	72 КБ
STDOLE2.TLB	06.10.1998 01:00	Файл "TLB"	18 КБ
VB6STKIT.DLL	18.06.1998 01:00	Расширение при...	101 КБ

Рис.9. Розпаковка файла

1. Перша помилка це в написанні слова Edit тобто орфографічна помилка.
2. Друга помилка це лишня клітинка в списку



3. Третя помилка відбувається, коли ми вводим число більше 9999 та прибираємо галочку в розділі Symbols а саме в Numeric та Alphabetic і натискаємо Add to List. Ми побачимо помилку номер 5



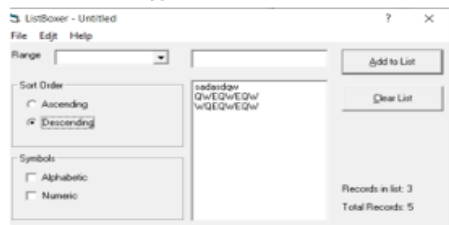
4. Четверта помилка це пуста клітинка в зоні Листа, тобто ми можемо виділити пусту зону, якої не повинно бути.



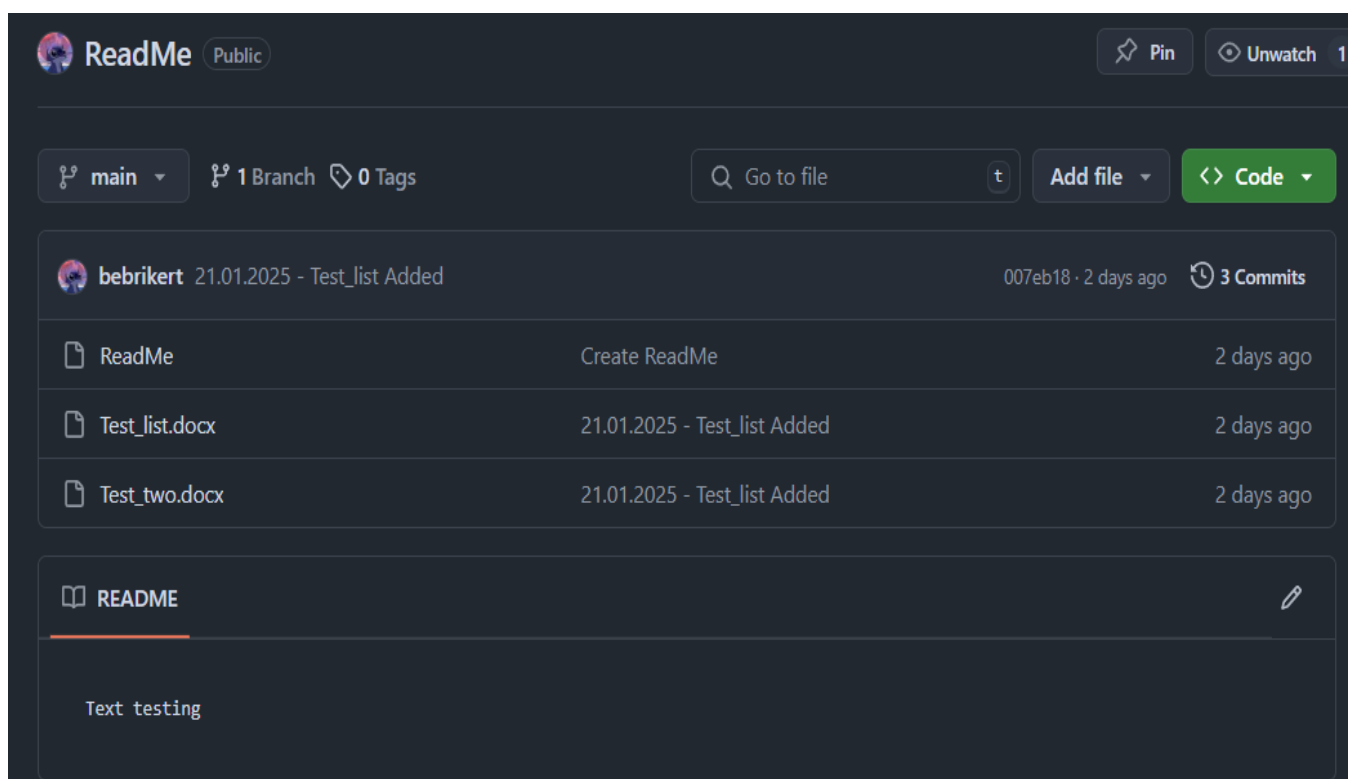
5. Остання помилка я побачив при написаннілюбих слів та виведення їх на лист можливо зловити баг. Коли ми пишемо наприклад **Сон** ми можемо перейти в розділ Symbols та прибрати галочки з цих



ми бачимо, що текст пропав, але якщо змінити один пункт в розділі **Soft Older** ми побачимо текст знову, але вже без пунктів які ми прописали зверху. Остаточний вигляд має так:



### Рис. 11. Помилки програми



**Рис. 12. Додано файл з помилками програми на репозиторій - <http://surl.li/ppnfbx>**



**Висновок:** Я став на деяку мить тестувальником, який знаходив баги та проблеми програми і намагався їх розписати, щоб їх виправили в майбутньому. Далі всі проблеми я завантажив на репозиторій, щоб всі користувачі, які бачили мої сторінку могли бачити самі помилки програми

### Завдання 5.

Створіть команду у Trello. Додайте тестову документацію до свого проекту. Оцінка роботи.

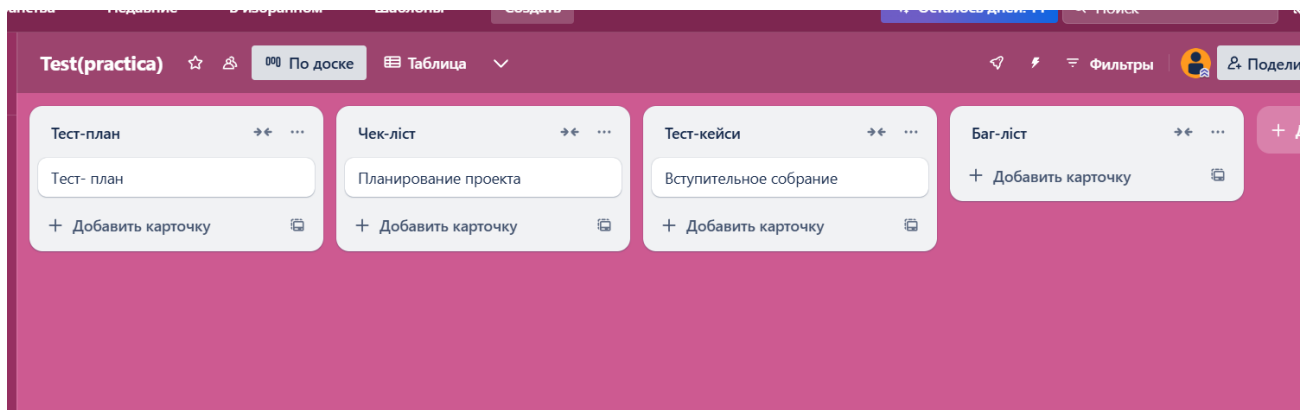


Рис.13. Дошка для роботи - <http://surl.li/smcrrfh>

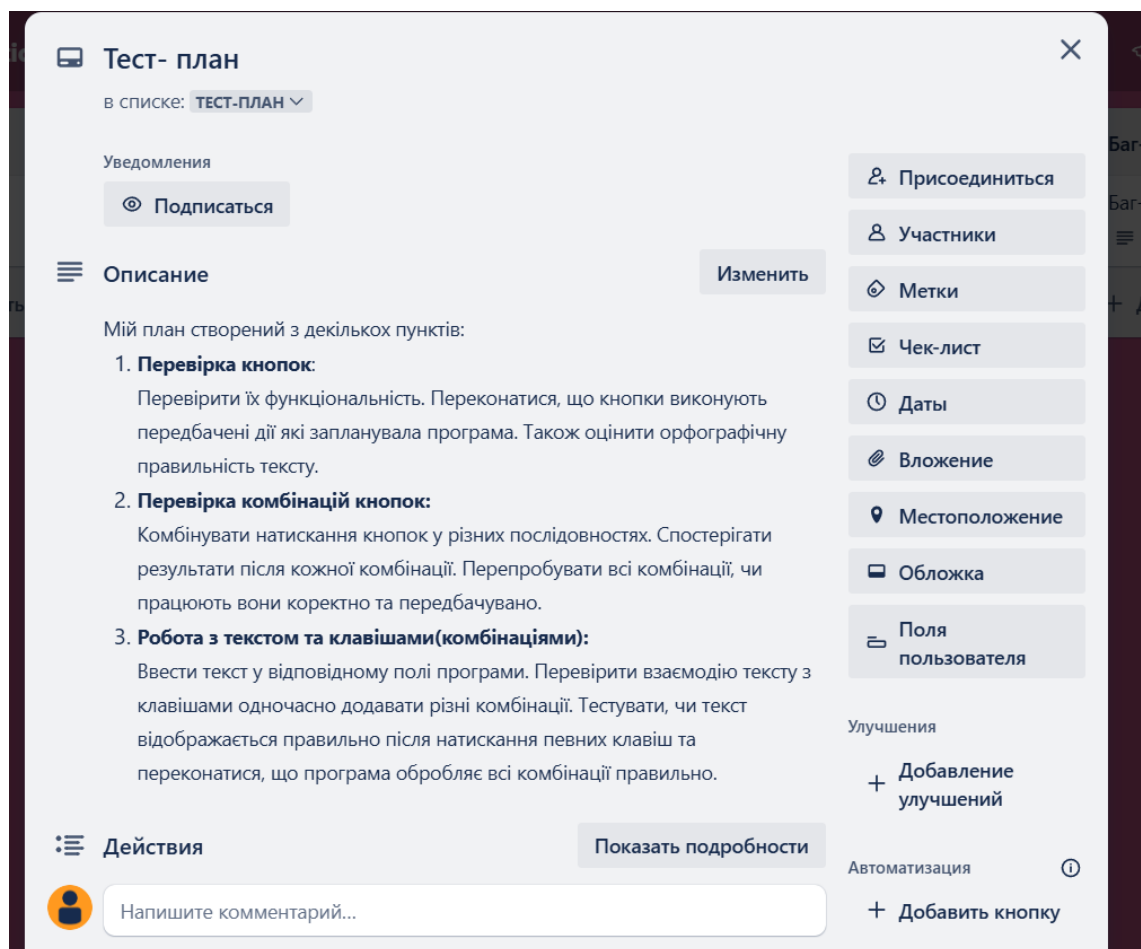


Рис.14. Опис колонки “Тест-план”

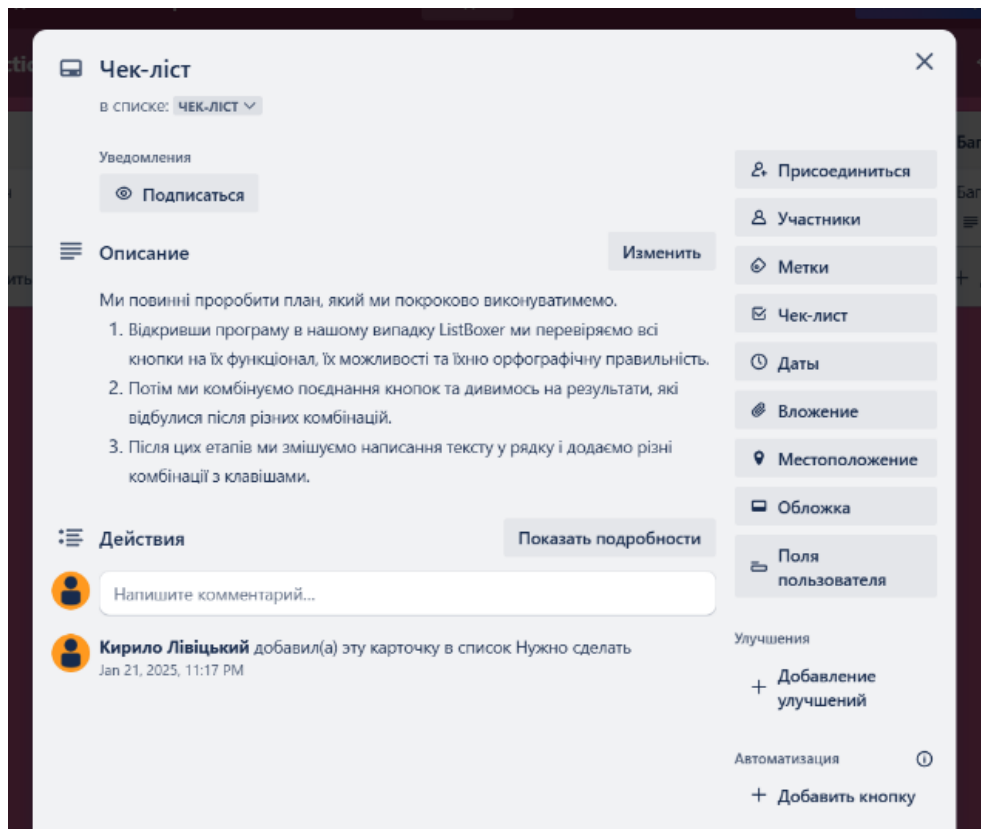


Рис.15. Опис колонки “Чек-ліст”

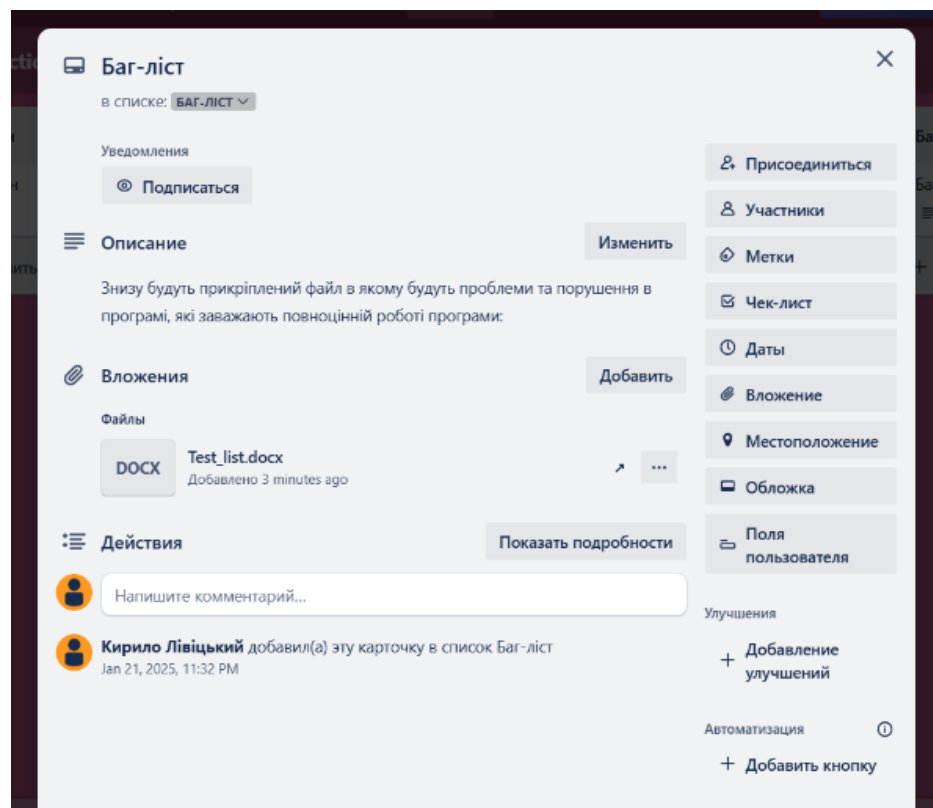


Рис.16. Опис колонки “Баг-ліст”

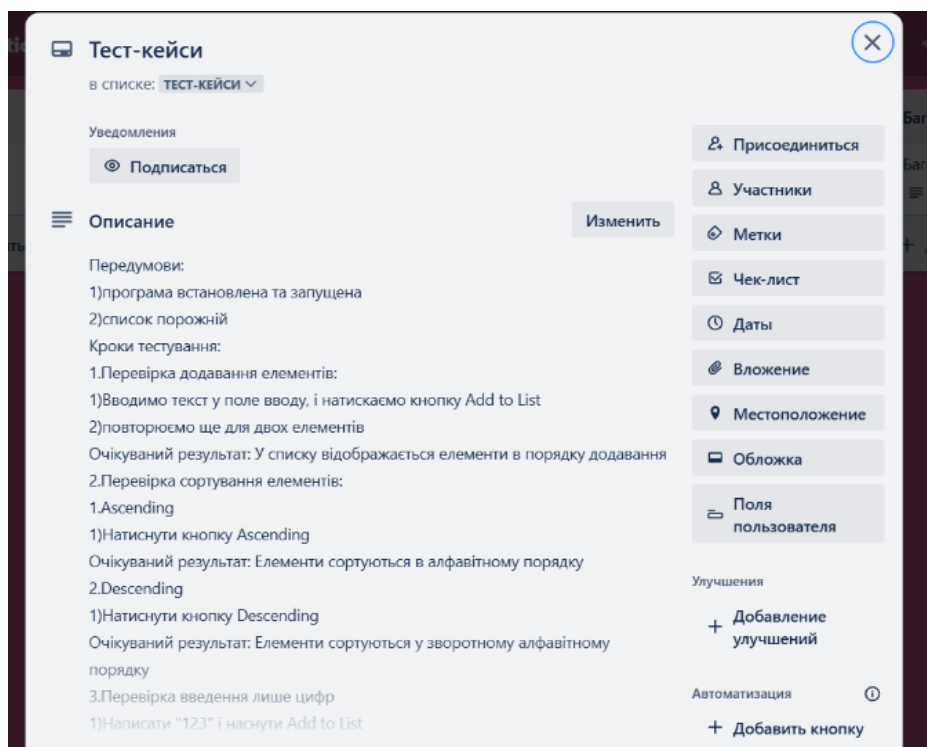


Рис.17. Опис колонки “Тест-кейси”

**Висновок:** Я вивчив для себе нову платформу Trello, яка мені допомогла в зберіганні даних та їх сортування по різних категоріях, щоб знаходження тої чи іншої інформації відбувалось в декілька раз швидше.

### Завдання 6.

**Частина 1.** Установка KODU GAME LAB. Створення тривимірної гри. Надішліть завдання на експертизу

**Частина 2.** Ознайомлення з дизайн-документом гри. Підготовка шаблону дизайн-документу. Створення сценарію гри та заповнення дизайн-документу. Громадська експертиза.

### Частина 1

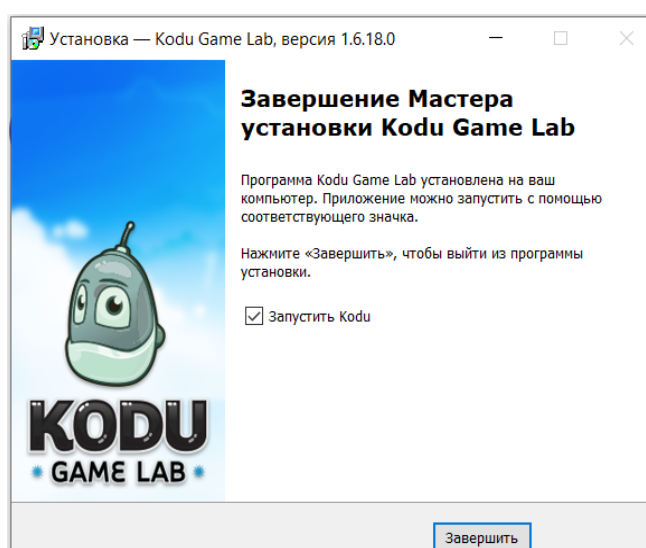
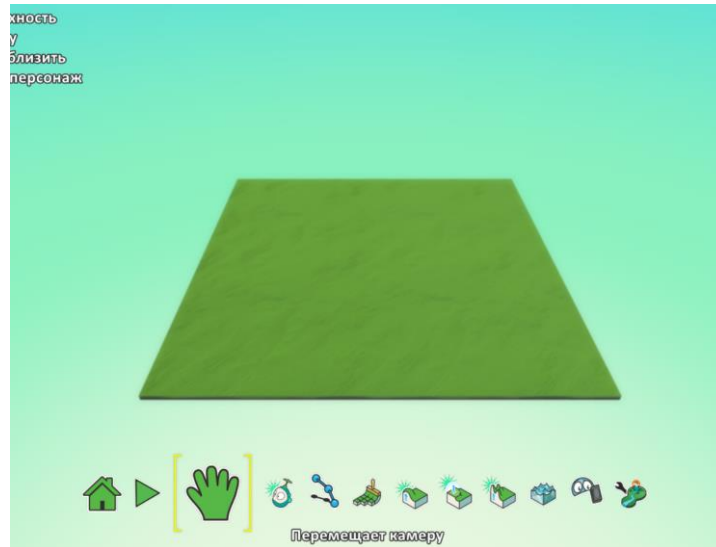


Рис.18. Завершення завантаження Code Game Lab



**Рис. 19. Створення порожнього світу для Гри 1**



**Рис. 20. Додавання об'єкта дерево**



**Рис. 21. Додавання об'єкта Байкер**

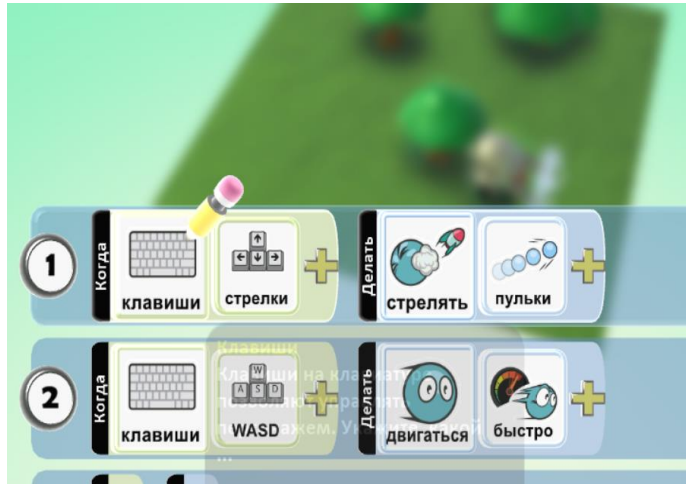


Рис. 22. Задавання програми байкеру

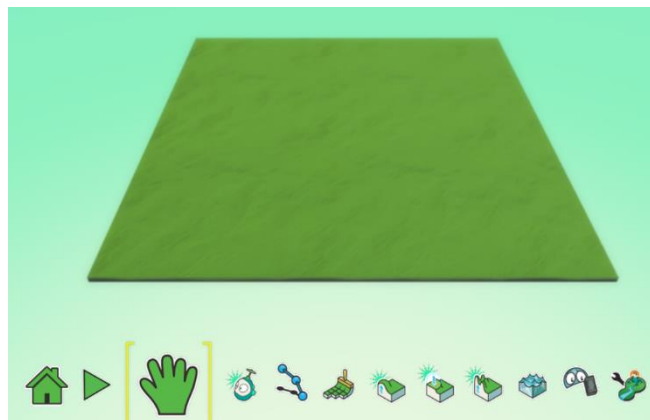


Рис. 23. Створення порожнього світу для Гри 2

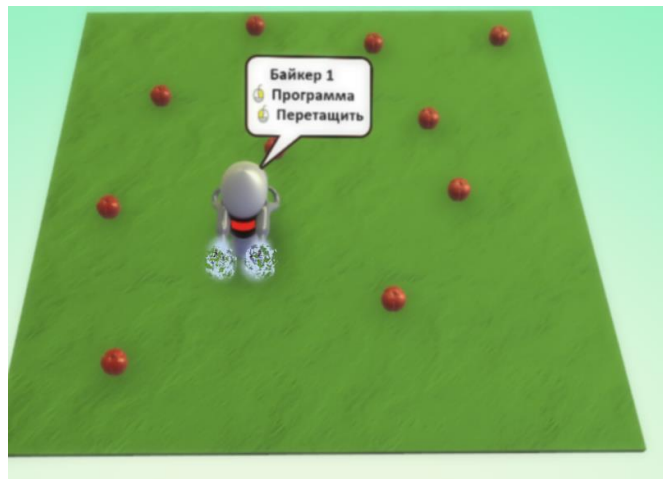


Рис. 24. Додавання об'єкта "Яблуко"



Рис. 25. Задавання програми байкеру

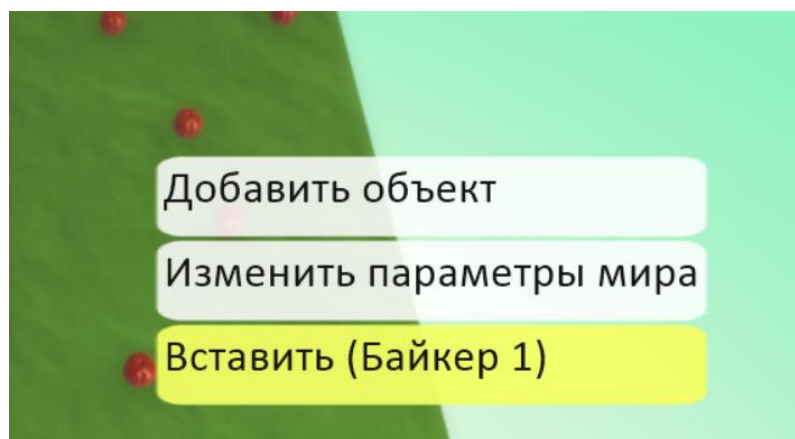


Рис. 26. Додано второго байкера



Рис. 27. Задавання програми другому байкеру

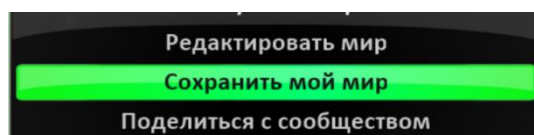


Рис. 28. Меню збереження світу

## Частина 2.

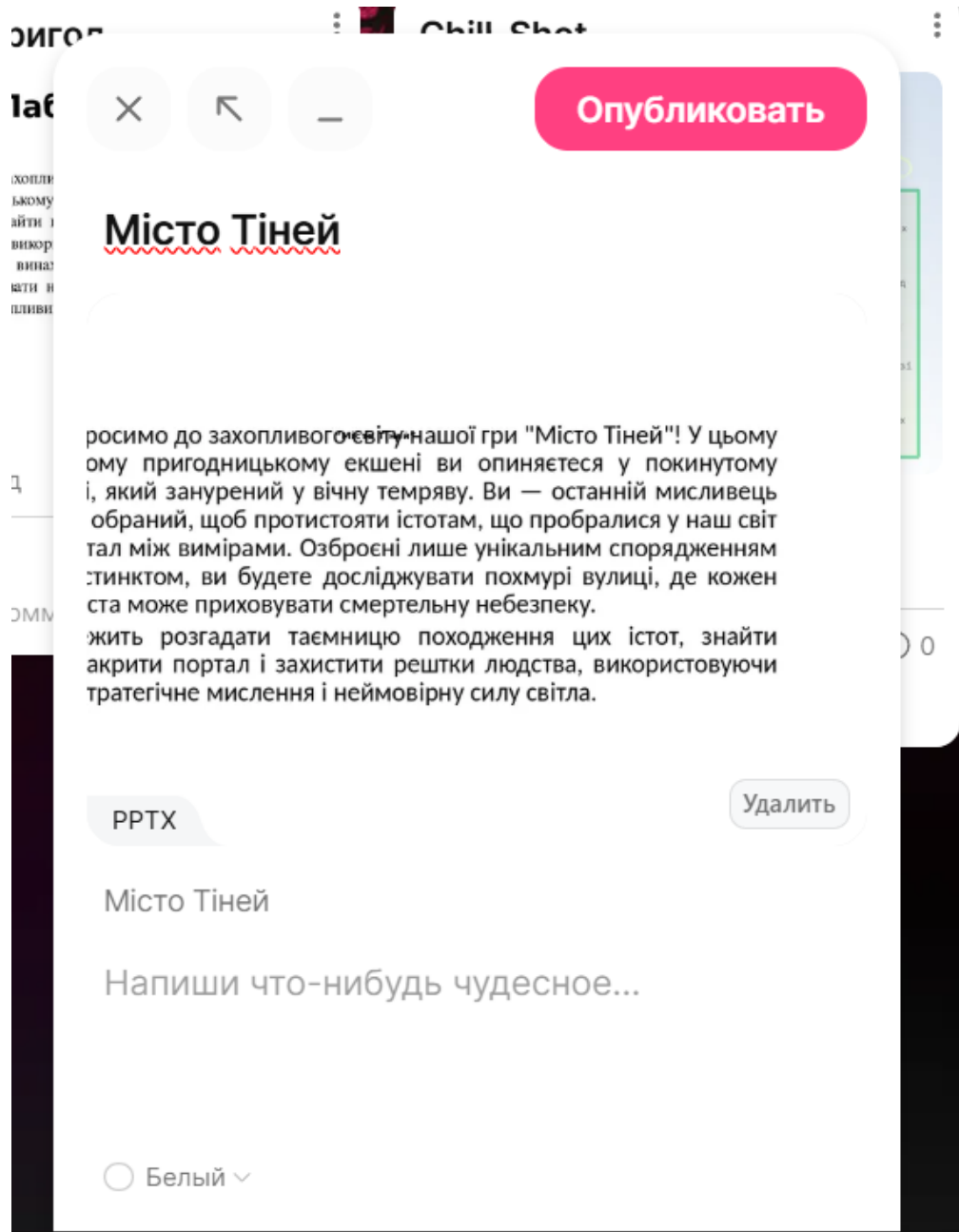


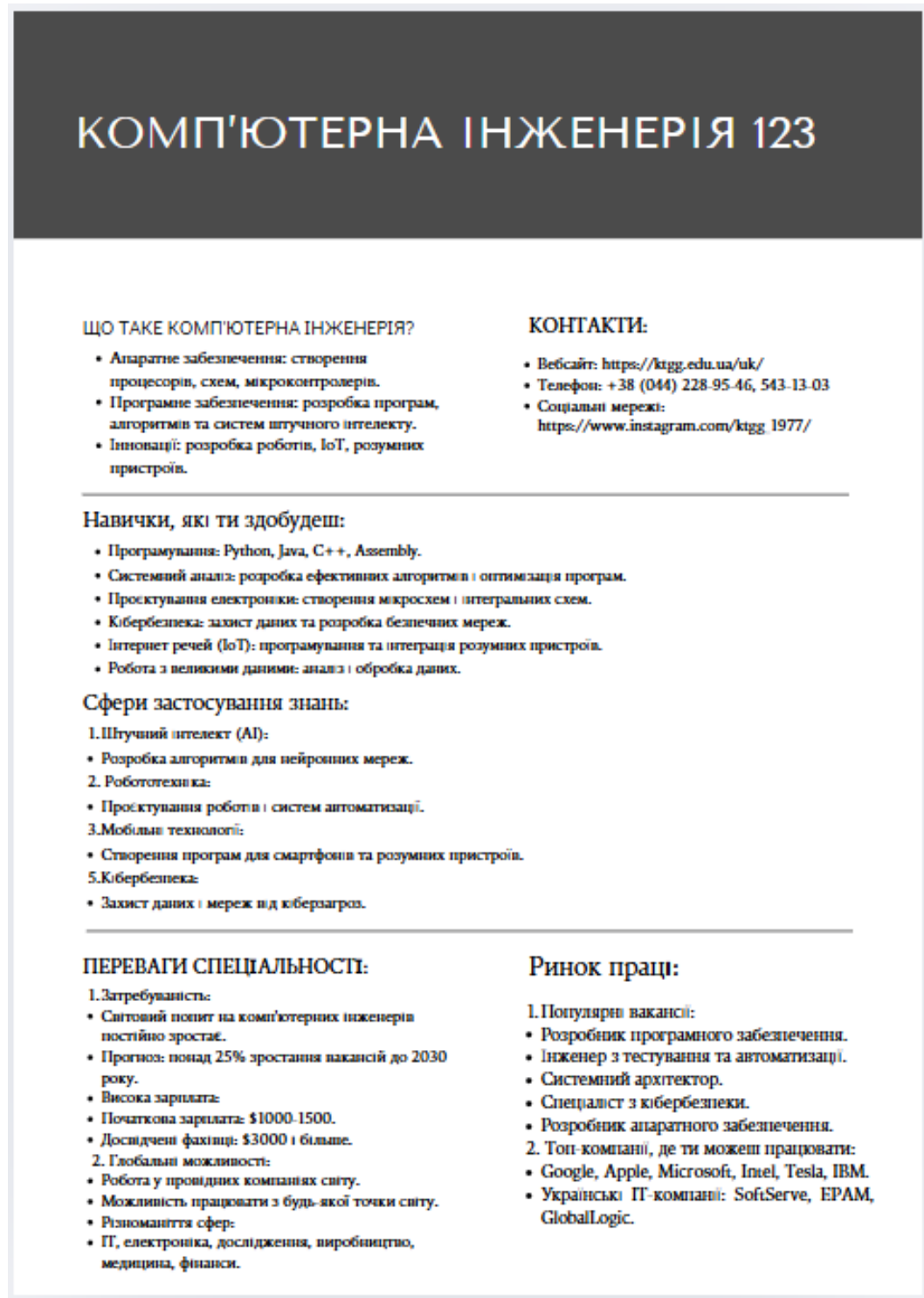
Рис. 29. Вікно додавання тексту

**Висновок:** Було проведено великий обсяг роботи з текстом та програмою. З самого важкого може виділити це завдання програмі об'єкту, в нашому випадку це був байкер він зміг рухатися як і самостійно так і за допомогою ведення клавіатури, також виконував роботи як викрик діалогу написаний мною та стрільба по об'єктах. Все інше, а точніше створення світу, додавання об'єктів та зміна кольорів об'єктам були не настільки складним, щоб з цим з'явилися труднощі.



**Завдання 7.**

Зареєструйтесь на сайті Canva. Створіть за допомогою сервісу Canva інфографіку для реклами своєї спеціальності "Інженерія програмного забезпечення" та "Комп'ютерна інженерія" відповідно. Оцінка роботи.



**Рис. 30. Оформлення інфографіки для реклами**



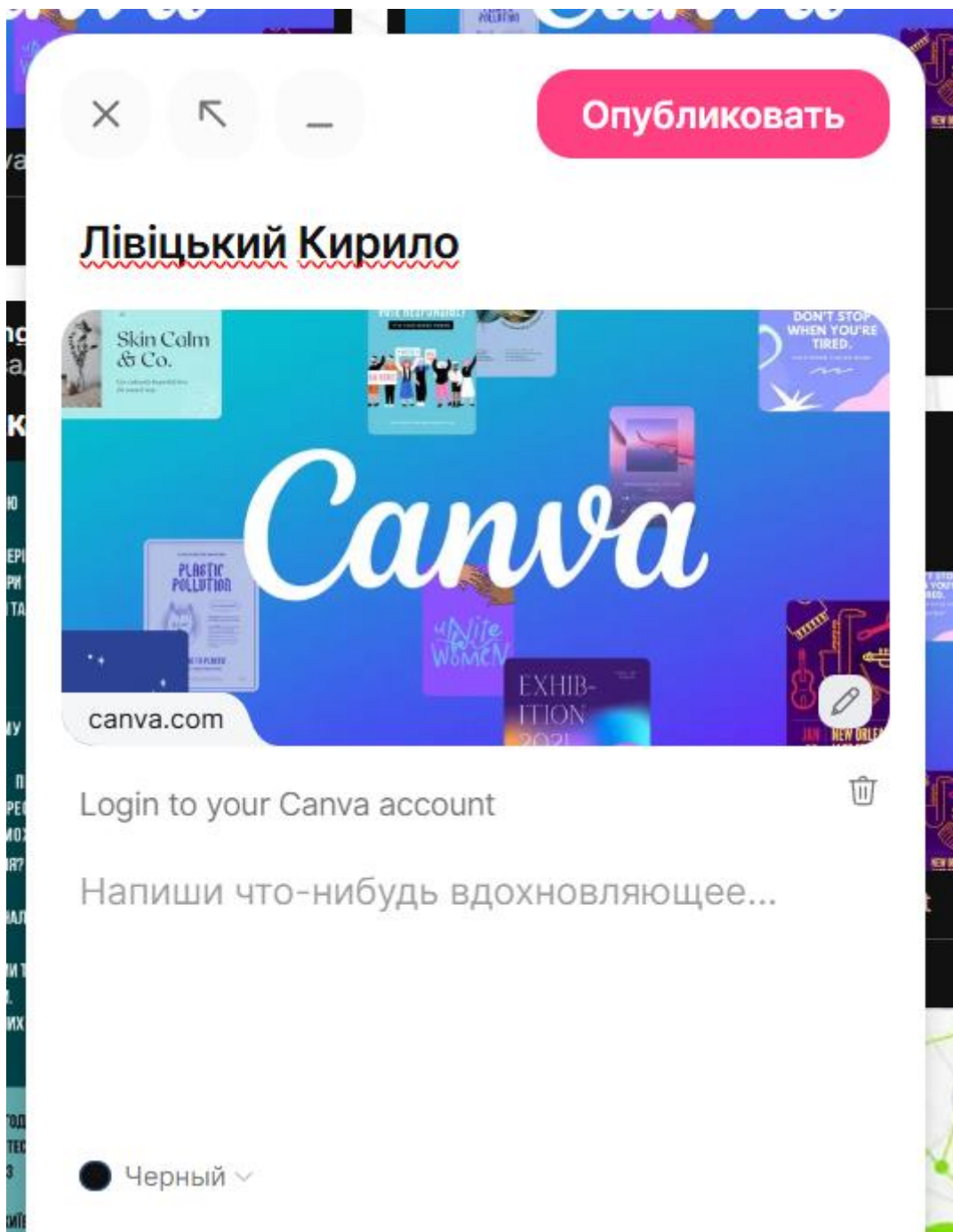
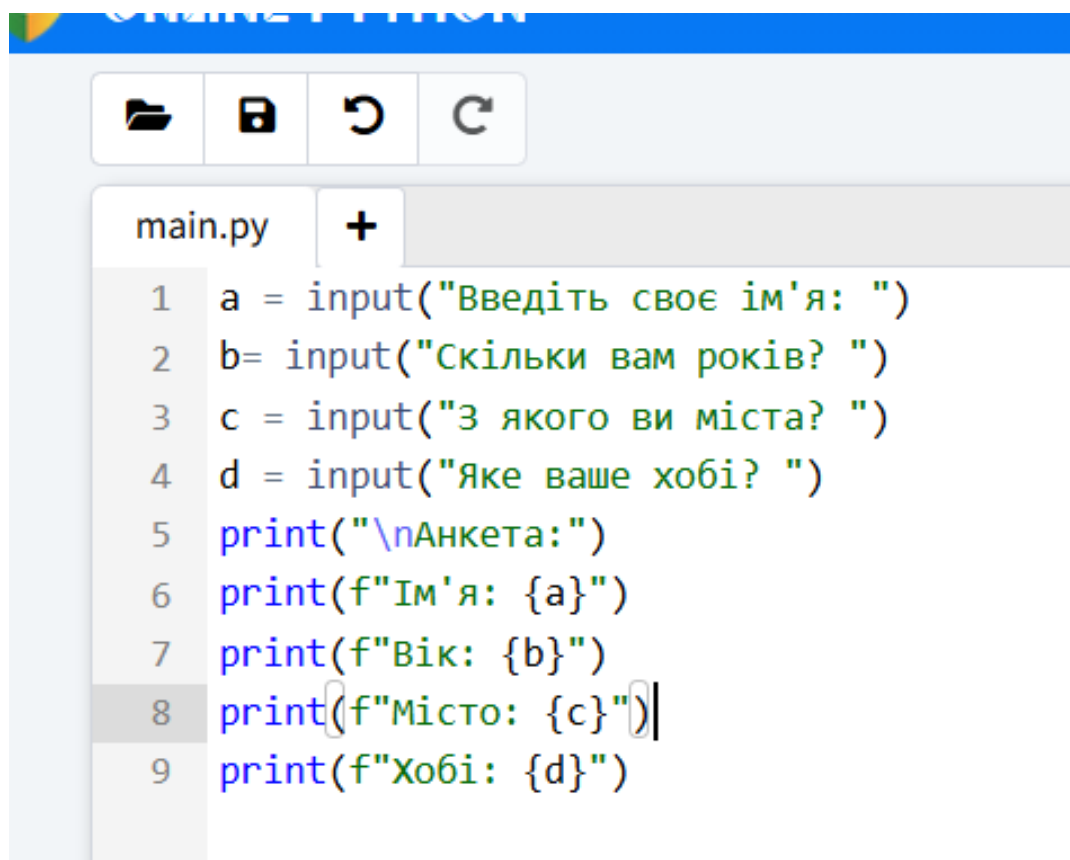


Рис. 31. Публікація реклами на дошці - <https://surl.li/iukywg>

**Висновок:** В даному завданні я навчився створювати інфографіку для реклами різних закладів, осіб, речей в залежності від прохання клієнта. Це допоможе навіть в майбутньому створити для себе біографію в такому вигляді та оформити її по всіх нормах та правилах.

**Завдання 8.**

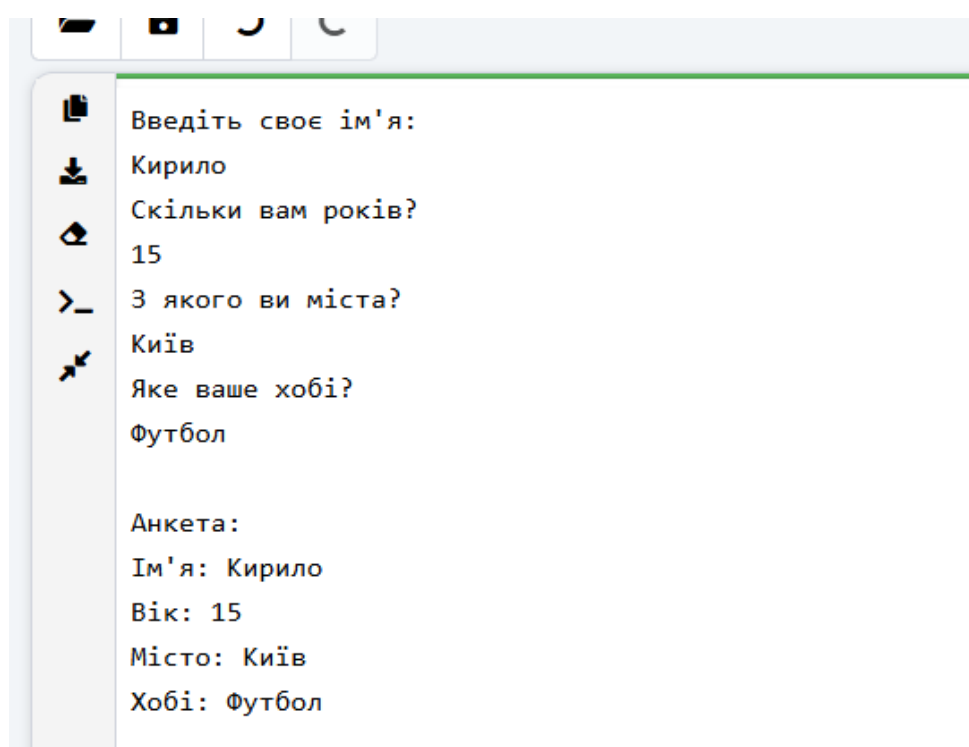
Відкрити сайт [Online Python - IDE, Editor, Compiler, Interpreter](https://onlinepython.net/). Виконання завдань. Виконайте самостійне завдання та перешліть на перевірку.



```

main.py
1  a = input("Введіть своє ім'я: ")
2  b= input("Скільки вам років? ")
3  c = input("З якого ви міста? ")
4  d = input("Яке ваше хобі? ")
5  print("\nАнкета:")
6  print(f"Ім'я: {a}")
7  print(f"Вік: {b}")
8  print(f"Місто: {c}")
9  print(f"Хобі: {d}")
  
```

**Рис.32. Написання коду – анкети**



```

Введіть своє ім'я:
Кирило
Скільки вам років?
15
З якого ви міста?
Київ
Яке ваше хобі?
Футбол

Анкета:
Ім'я: Кирило
Вік: 15
Місто: Київ
Хобі: Футбол
  
```

**Рис. 33. Приклад анкети, яка вивелась за допомогою коду**

**Висновок:** Я ознайомився з способом програмування через програму Python в якій я створив коротку анкету на яку може відповідати кожний, кому було надіслано цей код/файл.

## Робототехніка



**Рис. 34.** Збирання робота в команді

**Висновок:** Отримав новий досвід з робототехніки, коли я виконував цю роботу я намагався діяти в команді і допомагати один одному з різних питань від роз'яснення інструкції до підключення робота до мікросхеми.

## Загальний висновок

Отже, за весь час виконання завдань я отримав цінний досвід в різних аспектах тестувань. Перше, що я б хотів зачепити за час моїх досліджень це тестування олівця, а точніше це проведення smoke, функціональне, UX-тестування, тестування безпеки, стрес-тестування предмета, через які я дізнався, що олівець пройшов перевірки та виявився якісним, зручним і міцним, хоча ламкість грифеля при сильному натиску є типовим недоліком. Наступним завданням був тест “Фінансовий сенсей” який я успішно пройшов та дізнався багато цікавого про фінанси та їх використання різними способами без небезпечних наслідків. Третє завдання було початком мого шляху як фахівця, тобто реєстрація на сайті GitHub на якому я вже навчився як створювати репозиторій на якому можна тримати всю інформацію, тобто сховище кодів, програм, додатків. Наступний крок на моєму шляху це спроба себе як тестувальником програми в нашому випадку ця програма називалась ListBoxer. На цьому ж кроці я дізнався яка різниця між Test-Case, Test-Plan, Bag-report, Chek-list і як вони взаємодіють в різних аспектах між собою. При відкритті програми відразу було знайдено декілька багів, які я одразу записав в текстовий документ для того, щоб програмісти їх виправили в наступних оновленнях. Цей крок я гадаю був важливий,

щоб зрозуміти як тяжко виправити різні баги в різних програмах/ іграх і тд. Наступне завдання, яке я виконав впливає з попереднього, а точніше відсортування помилок, які я знайшов в різних розділах, а точніше їх роз'ясенням та як воно повинно показувати чи працювати без багу. Це завдання було складніше за попереднє, але я з ним справився. Наступним завданням це було створені гри в програмі KODU GAME LAB. Я рахую, що це завдання були на фантазію, як ти собі можеш уявити свою гру і що там буде відбуватись. Це завдання було зроблено, я навчився механіці програми KODU GAME LAB та як з нею працювати. Звісно це чудового, але друга частина була ще цікавіша тим, що нас самих просять створити сюжет та персонажів гри в моєму випадку я створив розповідь з назвою "Місто Тіней" та своїми уявними персонажами. Майже перед останнім завданням було створити інфограму для реклами своєї спеціальності, за допомогою платформи Canva через яку я знайшов дизайн та макет своєї реклами. Після цих дій я свою рекламу виставив на дошку для перегляду. І перед останнє завдання це я створив код – анкету через Python. Було використано такі команди як print, input та зміні в моєму випадку a,b,c,d за яких я зміг зробити свою власну анкету. І останній пункт і самий цікавий це робототехніка на якій я з командою склав свій перший робот, я дуже вдячний за взаємодопомогу та розуміння в цій сфері це було дуже доречно.