

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук та інформаційних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка телеграм помічника для онлайн навчальної платформи “ІТ курси”»

Студента 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

Матюшка Кирила
Костянтиновича

підпис студента

Науковий керівник
кандидат педагогічних наук, доцент

Дивак Володимир
Валерійович

підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____

Затверджую

Пурський О.І.

«12» грудня 2022р.

**Завдання
на випускну кваліфікаційну роботу (проект) студента**

Матюшка Кирила Костянтиновича

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка телеграм помічника для онлайн навчальної платформи "ІТ курси"»
Затверджена наказом ректора від «09» грудня 2022 р. № 3332
2. Строк здачі студентом закінченої роботи 30 травня 2023 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: розробити телеграм помічника для навчальної платформи з мов програмування
Об'єкт дослідження: процес розробки телеграм помічника для навчальної платформи з мов програмування
Предмет дослідження: засоби розробки телеграм помічника для навчальної платформи "ІТ курси"
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється

консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дивак В.В.	15.12.2022 р.	15.12.2022 р.
2	Дивак В.В.	15.12.2022 р.	15.12.2022 р.
3	Дивак В.В.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

1.1. Аналіз предметної області

1.2. Постановка задачі

1.3. Аналіз існуючих способів рішення задачі

РОЗДІЛ 2. МОДЕЛЬ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

2.1. Математична модель розрахунку часу для створення телеграм боту

2.2. Методи і технологія розробки телеграм боту

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

3.1. Програмна реалізація телеграм боту

3.2. Інструкція користувача телеграм боту

3.3. Тестування та перевірка телеграм боту

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	04.10.2022	04.10.2022
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2022	15.12.2022

3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<u>РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ</u> <u>ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ</u> <u>ПЛАТФОРМИ 3 МОВ ПРОГРАМУВАННЯ</u>	28.02.2023	28.02.2023
5	<u>РОЗДІЛ 2. МОДЕЛЬ ТЕЛЕГРАМ БОТУ</u> <u>ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ 3 МОВ</u> <u>ПРОГРАМУВАННЯ</u>	06.04.2023	06.04.2023
6	<u>РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ</u> <u>ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ</u> <u>ПЛАТФОРМИ 3 МОВ ПРОГРАМУВАННЯ</u>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	31.05.2023 -01.06.2023	31.05.2023 - 01.06.2023
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.06.2023	02.06.2023
12	<i>Представлення готової зшитой випускної кваліфікаційної роботи на кафедрі</i>	05.06.2023	05.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи (проекту)

Дивак В.В.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

Завдання прийняв студент-дипломник

Матюшок К.К.

(прізвище, ініціали, підпис)

9. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

30.05.2023 р.

(підпис, дата)

10. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента _____

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми _____

Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри _____

Пурський О.І.

(підпис, прізвище, ініціали)

« _____ » 2023 р.

Анотація

Випускна кваліфікаційна робота присвячена розробці та реалізації інноваційного рішення для поліпшення процесу навчання на онлайн платформі "ІТ курси". У цій роботі досліджуються можливості і переваги використання месенджера Telegram як додаткового інструменту для забезпечення зручного та ефективного навчання, проводиться аналіз потреб і вимог користувачів платформи "ІТ курси" та досліджує функціональні можливості Telegram. На основі цього аналізу розробляється та реалізується Telegram-бот, який допомагає здійснювати різні навчальні операції, такі як доступ до курсів, сповіщення про нові матеріали, виконання завдань, отримання допомоги в розв'язанні завдань тощо.

Ключові слова: telegram, інструмент для навчання, аналіз потреб та вимог, інформаційна технологія.

Anotation

The graduation to develop and implement innovative solutions, theses are to enhance the learning process on the online platform 'IT Courses.' This work explores the possibilities and advantages of using the Telegram messenger as an additional tool for facilitating convenient and effective learning. It conducts an analysis of the needs and requirements of the 'IT Courses' platform users and investigates the functional capabilities of Telegram. Based on this analysis, a Telegram bot is developed and implemented, which assists in various learning operations such as accessing courses, receiving notifications about new materials, completing tasks, and obtaining assistance in task solving.

Keywords: telegram, learning tool, needs and requirements analysis, information technology."

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ	9
1.1. Аналіз предметної області	9
1.2. Постановка задачі.....	9
1.3. Аналіз існуючих способів рішення задачі	10
РОЗДІЛ 2. МОДЕЛЬ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ	12
2.1. Математична модель розрахунку часу для створення телеграм боту .	12
2.2. Методи і технологія розробки телеграм боту.....	17
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ	20
3.1. Програмна реалізація телеграм боту	20
3.2. Інструкція користувача телеграм боту.....	24
3.3. Тестування та перевірка телеграм боту	27
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
ДОДАТОК	32

ВСТУП

Студент стикається з такою проблемою, як розряджений телефон, та невелика кількість мобільного трафіку, тому було вирішено створити бота в зручному та популярному месенджері під назвою Telegram. За допомогою цього бота можна зменшити використання трафіку та збільшити тривалість життя смартфона.

На сьогоднішній день існують багато телеграм ботів, більшість з яких були створені в спеціальних конструкторах. Але зазвичай вони не є зручними, та мають свої недоліки. Такі як перевантажений функціонал, низька швидкість реагування бота, незручне та незрозуміле меню з безліччю гілок іншого меню.

Метою дипломної роботи є створення бота для навчальної платформи з мов програмування.

Для досягнення мети були поставлені наступні цілі:

1. Провести аналіз сучасних систем розробки телеграм ботів;
2. Вивчити поняття телеграм бота, суть, функції, класифікацію;
3. Виконати проектування телеграм бота;
4. Розробити телеграм бота для сайту.

Об'єктом дослідження є телеграм бот.

Предмет дослідження – розробка телеграм помічника для навчальної платформи “ІТ курси”.

Практична цінність роботи полягає в тому, що в її основні положення, висновки та результати дають змогу:

- забезпечити ефективну роботу телеграм бота;
- більше розповісти про навчальну платформу;
- поглибити теоретичні та практичні знання мов програмування.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ПИТАННЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

1.1. Аналіз предметної області

Головний відділ такої платформи як наша – це відділ онлайн. У цьому відділі співробітники займаються розробкою та додаванням нових функцій, відповідей та тестів до платформи. Основне завдання, яке на даний час виконує компанія, це оптимізація та поліпшення роботи клієнта з ботом. Але для стабільного виконання навіть такої легкої задачі використовується значна кількість працівників які розділенні по відповідним відділам, а саме:

- Відділ спеціалістів, це програмісти з навичками, які оптимізують та поліпшують роботу з ботом.
- Відділ фінансів проводить підрахунок зароблених коштів та розподіляє заробітну плату.

1.2. Постановка задачі

Було вирішено розробити телеграм бота, який виконує навігацію на навчальній платформі по вивченню мов програмування.

Цей бот направлений на швидку роботу з сайтом не знаходячись на ньому.

Метою дипломної роботи є створення бота для навчальної платформи з мов програмування.

Завдання роботи оптимізація часу.

Бот повинен надавати зрозуміле меню та підказки.

Вхідна інформація:

Користувач:

- Повідомлення від користувача.
- Обрані кнопки в меню.
- Ім'я користувача.

Вихідна інформація:

- Посилання на сторінки сайту.
- Кількість пройдених тестів.
- Кількість пройдених тестів у відсотковому відношенні.
- Сторінки з тестами.

1.3. Аналіз існуючих способів рішення задачі

Переглянувши декілька платформ з навчальних курсів, не було знайдено таких ботів, які б полегшували роботу з сайтом. Зазвичай клієнти інших сервісів працюють напряду з сайтом, тому було вирішено розробити унікального телеграм помічника для швидкої роботи та навігації по сайту. Загалом, однією з важливих проблем багатьох навчальних сайтів є велика кількість інформації та посилань, у яких користувач може загубитися, для запобігання таких ситуацій і був зроблений телеграм помічник, який не тільки зменшить використання трафіку смартфона, але і полегшить “спілкування” користувача з сервісом. Поки що, телеграм помічник знаходиться в бета-тестуванні та виконує функцію навігації по сайту з невеличким тестом, але у майбутньому, з наступними оновленнями, його

функціональність буде тільки рости. Навіть зараз ведуться тестування оновлених функцій, як для користувача, так і для фірми в цілому.



РОЗДІЛ 2.

МОДЕЛЬ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

2.1. Математична модель розрахунку часу для створення телеграм боту

Загальний час на створення програми складається з різних компонентів. Структура загального часу на створення програмного продукту представлена в таблиці 3.1

Таблиця 3.1 – Структура загального часу на створення програмного забезпечення.

№ етапу	Позначення часу для цього кроку	Зміст етапів
1	$T_{по}$	Підготовка описів завдання
2	$T_{о}$	Опис завдання
3	$T_{а}$	Розробка алгоритму
4	$T_{бс}$	Розробка блок-схеми алгоритму
5	$T_{пн}$	Розробка інтерфейсу
6	$T_{н}$	Написання програм (кодування)
7	$T_{вт}$	Відладка та тестування програм
8	$T_{д}$	Документація, посібники користувача та інструкції

Час розраховується в люд-год, причому $T_{по}$ та $T_{д}$ береться по фактично відпрацьованому часу, а час останніх етапів визначається розрахунковий по умовному числу команд Q .

Умовне число команд Q визначається за формулою

$$Q = q * z, \quad (3.1)$$

де q - коефіцієнт, який враховує умовне число команд залежно від типу завдання. Орієнтовне значення коефіцієнта q вибирається з таблиці 3.2., але студент може самостійно обирати не з таблиці, виходячи із кількості рядків кодування та особливостей створення ПП.

Значення коефіцієнта q вибирається з таблиці 3.3.

Таблиця 3.2 – Значення коефіцієнта q .

Тип завдання	Значення коефіцієнта q
Завдання обліку	1400...1500
Завдання оперативного управління	1500...1700
Завдання планування	3000...3500
Багатоваріантні завдання	4500...5000
Комплексні завдання	5000...5500

Для даного завдання приймається коефіцієнт $q = 1650$.

Програмні продукти за мірою новизни можуть бути віднесені до однієї з чотирьох груп:

- група А - розробка принципово нових завдань;
- група Б - розробка оригінальних програм;
- група В - розробка програм з використанням типових рішень;

- група Г - разове типове завдання.

Для даного завдання міра новизни: А

За складністю програмні продукти можуть бути віднесені до однієї з трьох груп:

- 1 - алгоритми оптимізації і моделювання систем;

- 2 - завдання обліку, звітності і статистики;

- 3 - стандартні алгоритми.

Дане завдання може бути віднесено до другої групи складності низького рівня.

Коефіцієнт (з) визначається з таблиці 3.3 в залежності від складності і міри новизни.

Таблиця 3.3 – Значення коефіцієнта з.

Мова програмування	Група складності	Міра новизни			
		А	Б	В	Г
Високого рівня	1	1.38	1.26	1.15	0.69
	2	1.30	1.19	1.08	0.65
	3	1.20	1.10	1.00	0.60
Низького рівня	1	1.58	1.45	1.32	0.79
	2	1.49	1.37	1.24	0.74
	3	1.38	1.26	1.15	0.69

У цьому завданні коефіцієнт з = 1.38.

За формулою 3.1 визначається умовне число команд Q.

$$Q = 1650 * 1.38 = 2277 \text{ чол / год.}$$

Визначається час, витрачений на кожен етап створення програмного продукту:

- $T_{по}$ (час на підготовку опису завдання), береться по факту і для даного завдання

$$T_{по} = 40 \text{ чол / год.}$$

- T_o (час опису завдання) визначається за формулою

$$T_o = Q * B / (50 * K), \quad (3.2)$$

де B — коефіцієнт, що враховує зміну завдання. Коефіцієнт B залежно від складності завдання і числа змін вибирається в інтервалі від 1,2 до 1,5. У цьому завданні $B = 1,4$

K - коефіцієнт, який враховує якості програміста. Значення коефіцієнта вибирається з таблиці 3.4.

Таблиця 3.4 – Значення коефіцієнта K.

Стаж програміста	Значення коефіцієнта K
до 2-х років	0,8
від 2 до 3 років	1,0
від 3 до 5 років	1,1...1,2
від 5 до 10 років	1,2...1,3

понад 10 років	1,3...1,5
----------------	-----------

В даному випадку коефіцієнт $K = 1,2$.

Формула 3.2 обчислює час на опис завдання.

$$T_0 = 2277 * 1,4 / (50 * 1,2) = 53,13 \text{ [чол. / годин]}.$$

- T_A (час на розробку алгоритмів) розраховується за формулою

$$T_A = Q / (50 * K), \quad (3.3)$$

Формула 3.3 обчислює час на розробку алгоритмів.

$$T_A = 2277 / (50 * 1,2) = 37,95 \text{ [чол. / годин]}.$$

- T_{BC} (час на розробку блок-схеми) визначається аналогічно T_A за формулою 3.3 і складає

$$T_{BC} = 2277 / (50 * 1,2) = 37,95 \text{ [чол. / годин]}.$$

- T_H (час написання програми на мові програмування) визначається формулою

$$T_H = Q * 1,5 / (50 * K), \quad (3.4)$$

Формула 3.4 обчислює час, необхідний для написання програми.

$$T_H = 2277 * 1,5 / (50 * 1,2) = 56,93 \text{ [чол. / годин]}.$$

- T_{PI} (час на проектування інтерфейсу) визначається за формулою

$$T_{PI} = Q / 50, \quad (3.5)$$

Формула 3.5 обчислює час на розробку інтерфейсу програми.

$$T_{\text{П}} = 2277 / 50 = 45,54 \text{ [чол. / годин]}.$$

- $T_{\text{ВТ}}$ (час відладки і тестування програми) визначається за формулою

$$T_{\text{ВТ}} = Q * 4,2 / (50 * K), \quad (3.6)$$

Формула 3.6 обчислює час відладки і тестування програми.

$$T_{\text{ВТ}} = 2277 * 4,2 / (50 * 1,2) = 159,39 \text{ [чол. / годин]}.$$

- $T_{\text{Д}}$ (час на оформлення документації), береться по факту і для даного завдання

$$T_{\text{Д}} = 40 \text{ чол. / годин.}$$

Підраховується загальний час на створення програмного продукту за формулою

$$T = T_{\text{ПО}} + T_{\text{О}} + T_{\text{А}} + T_{\text{БС}} + T_{\text{Н}} + T_{\text{П}} + T_{\text{ВТ}} + T_{\text{Д}}, \quad (3.7)$$

$$T = 40 + 53,13 + 37,95 + 37,95 + 56,93 + 45,54 + 159,39 + 40 = 470,89 \text{ [чол. / годин]}.$$

2.2. Методи і технологія розробки телеграм боту

При розробці дипломного проекту була використана мова програмування Python.

Python - це високорівнева об'єктно-орієнтована мова програмування загального призначення з відкритим кодом. Її розробив Гвідо ван Россум у 1990 році. Ці високорівневі структури даних разом із динамічною семантикою та динамічним зв'язуванням є привабливими для швидкої розробки додатків і як спосіб об'єднання існуючих компонентів Python

підтримує модулі та модульні пакети, що полегшує модуляризацію та повторне використання коду Python підтримує модулі та модульні пакети, що полегшує модуляризацію та повторне використання коду. На всіх основних платформах інтерпретатор і стандартна бібліотека Python доступні як у скомпільованому, так і у вихідному форматі. Мова програмування Python підтримує декілька парадигм програмування, включаючи об'єктно-орієнтовану, процедурну, функціональну та аспектно-орієнтовану.

Інструментальні засоби розробки програмного продукту. Для створення боту використано Visual Studio Code - середовище розробки.

VS Code - це інструмент для створення, редагування та вилучення сучасних веб- і хмарних додатків. Microsoft анонсувала Visual Studio Code на конференції Build 2015 у квітні того ж року. Visual Studio Code базується на розробках вільного проекту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code - це надбудова над Atom Shell, яка використовує движок браузера Chromium та Node.js. На сайті Visual Studio Code, у прес-релізі та офіційному блозі не згадується про використання вільного проекту Atom. Редактор містить вбудований відладчик, засоби інтеграції з Git, рефакторингу, навігації по коду, автодоповнення типових конструкцій та контекстну довідку. Продукт підтримує розробку на платформах ASP.NET та Node.js і позиціонується як легке рішення, яке можна використовувати без повністю інтегрованого середовища розробки. Підтримувані мови та технології: JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS і SASS, Haxe.

Характеристика мови програмування. Загалом у проєкті використовується мова Python.

Мова Python є високорівневою мовою широкого застосування. Автор мови (Гвідо ван Россум) зробив сильний акцент на читабельності та синтаксисі коду. Мова дозволяє програмістам реалізовувати завдання зі значно меншою кількістю рядків коду, ніж вимагають інші мови, такі як C++ та Java.

Мова є кросплатформенною і може бути використана для різноманітних напрямків.

Мова підтримує парадигми, поширені в інших мовах програмування, такі як ООП, імперативне програмування та функціональне програмування. Python є динамічно типізованим. Він включає автоматичне керування пам'яттю та надає багато додаткових бібліотек.

Веб-розробка – це, мабуть, найпопулярніший напрямок для Python в даний час.

Мовою легко і швидко писати програми, що робить її ідеальним інструментом для швидкої розробки веб-сайтів. Крім того, набір існуючих веб-фреймворків та CMS-систем дозволяє створювати високоякісні веб-сайти за годин/дні: Django, Pyramid, Flask, Turbogears, Django-CMS, Plone CMS, Zope та багато інших...

РОЗДІЛ 3.

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ БОТУ ДЛЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ З МОВ ПРОГРАМУВАННЯ

3.1. Програмна реалізація телеграм боту

Функціональне призначення

Було вирішено розробити телеграм бота, який виконує навігацію на навчальній платформі по вивченню мов програмування.

Цей бот направлений на швидку роботу з сайтом не знаходячись на ньому



Завдання роботи оптимізація часу.

Бот повинен надавати зрозуміле меню та підказки.

Студент стикається з такою проблемою як розряджений телефон, та невелика кількість мобільного трафіку, тому було вирішено створити бота в зручному та популярному месенджері під назвою Telegram. За допомогою цього бота можна зменшити використання трафіку та збільшити тривалість життя смартфона.

Вхідні та вихідні дані. Вхідна інформація:

Користувач:

- Повідомлення від користувача.
- Обрані кнопки в меню.
- Ім'я користувача.

Вихідна інформація:

- Посилання на сторінки сайту.
- Кількість пройдених тестів.
- Кількість пройдених тестів у відсотковому відношенні.
- Сторінки з тестами.

Логічна структура програми та її модулів. У загальному представленні код розділений на 4 окремі частини. «Handlers» це головний файл, який до якого підключаються інші файли. У ньому знаходиться основний код.

Наступним є файл із назвою «keyboards» у якому знаходяться усі клавіатури. Далі є файл із назвою «config» у якому прописан Token телеграм бота. Останній файл це «bot», він потрібен для збереження Token та використання його у майбутніх файлах.

Організація інтерфейсу з користувачем. Інтерфейс інтуїтивно простий для будь якого користувача, завдяки зручним кнопкам, які автоматично з'являються на екрані після натискання кнопки.(рис1)

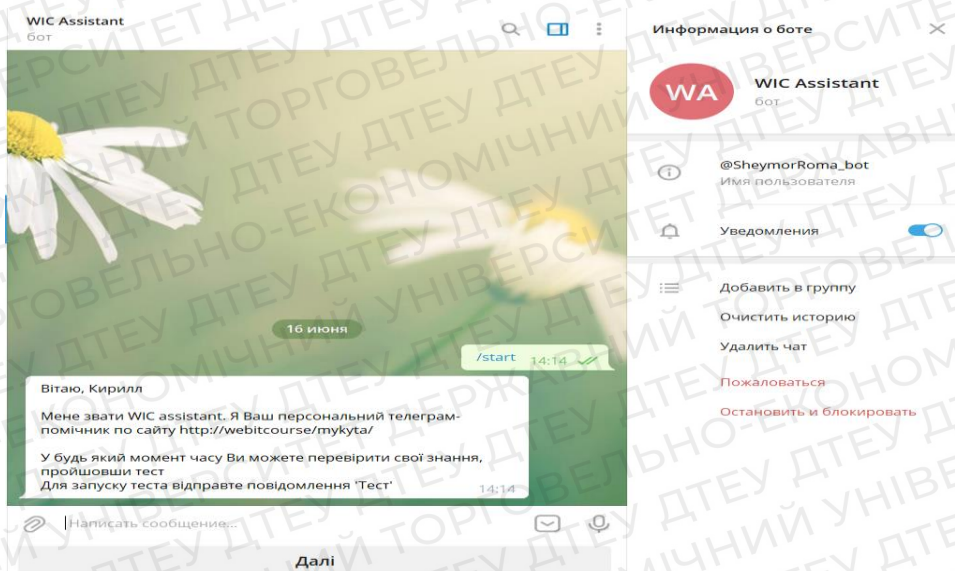
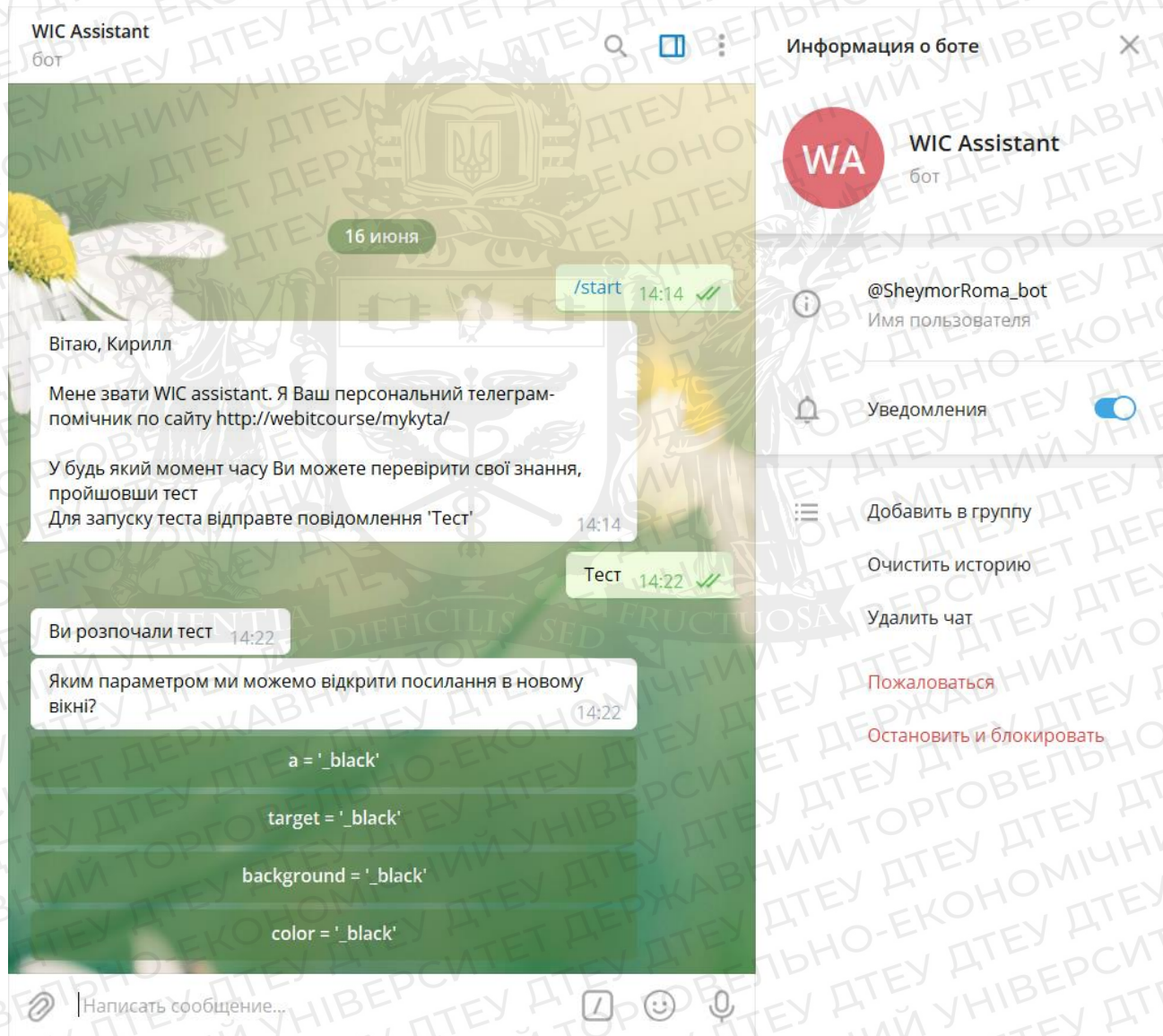


Рис 1. Інтерфейс телеграм бота

Також у ньому реалізований тест, який кожен користувач може почати виконувати у будь-який час, не втрачаючи попередньої кнопки. (рис 2 та рис 3)



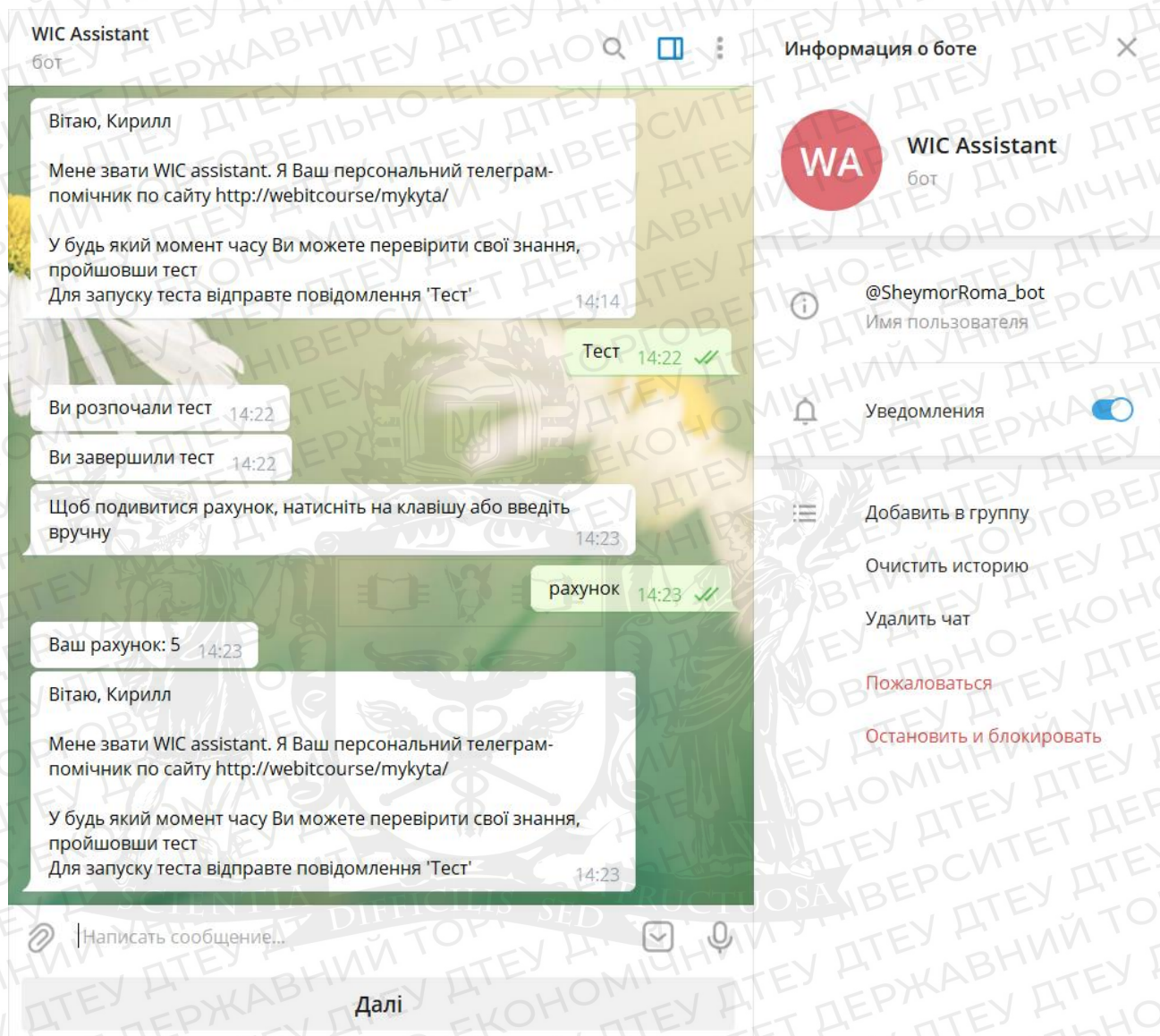


Рис 2 та 3. Тест

Характеристика технічних засобів. Онлайн платформа розроблена під пристрої які мають телеграм на ньому. Підійдуть такі пристрої, як смартфон, ноутбук, комп'ютер, планшет та інші.

3.2. Інструкція користувача телеграм боту

Вводиться команда “/start” після чого йде простий діалог з клавіатурою, яку може використовувати користувач. (рис 1)

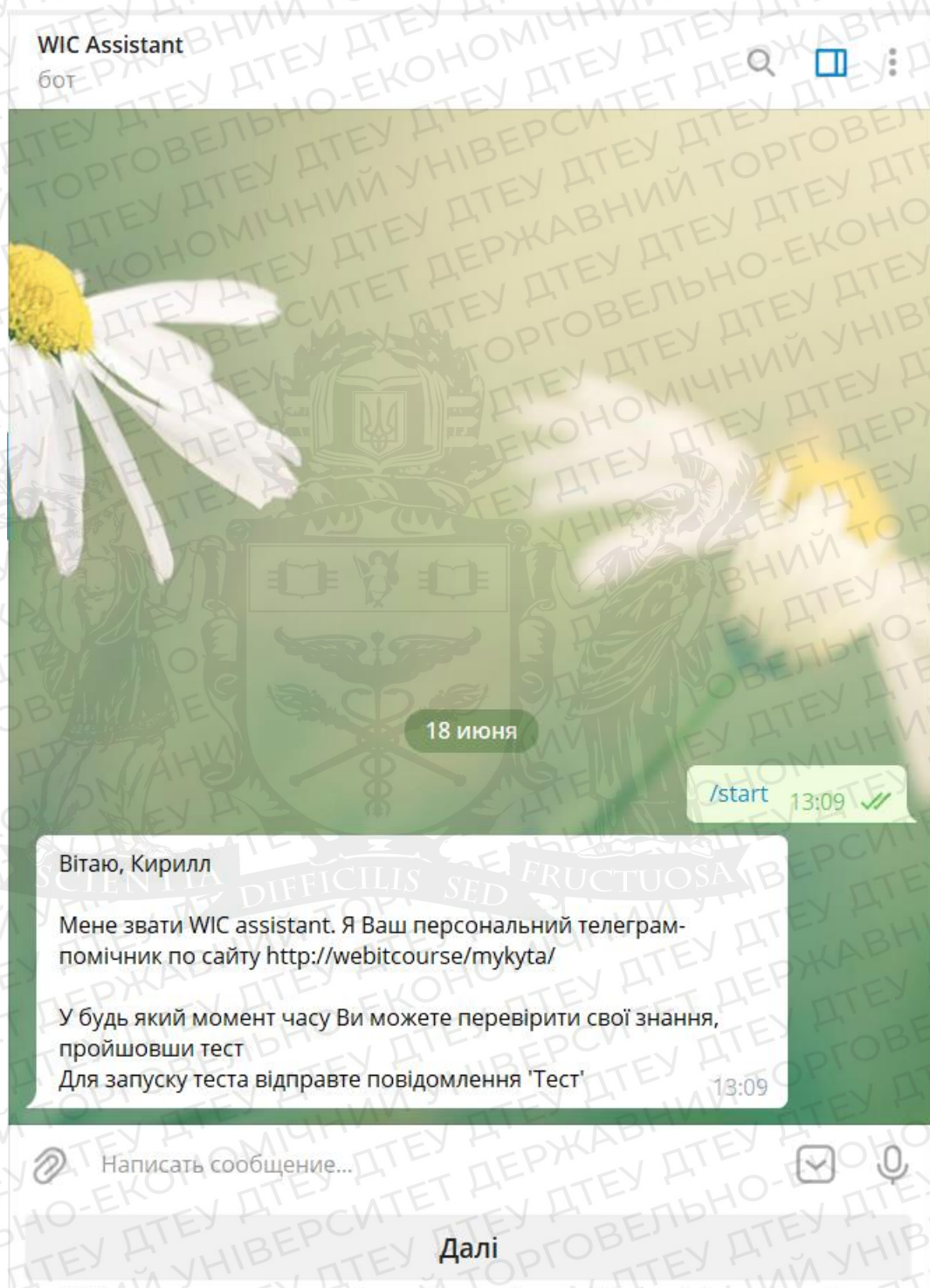


Рис 1. Початок роботи

У боті реалізован тест, для його початку потрібно ввести “Тест” (рис 2)

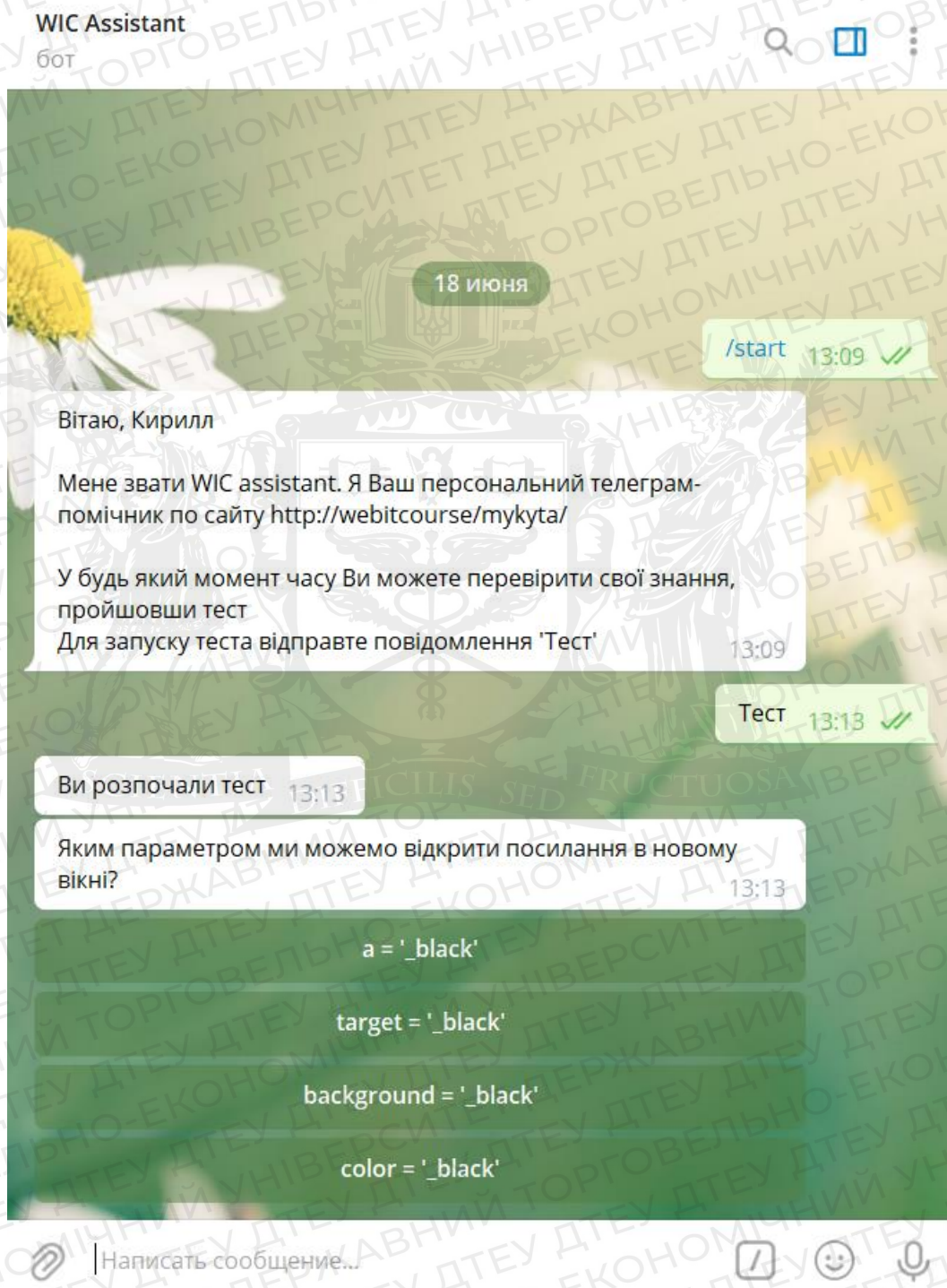


Рис 2. Початок тесту

3.3. Тестування та перевірка телеграм боту

Під час використання телеграм боту, користувач не може відправляти повідомлення боту, які в ньому не прописані. (рис 1)

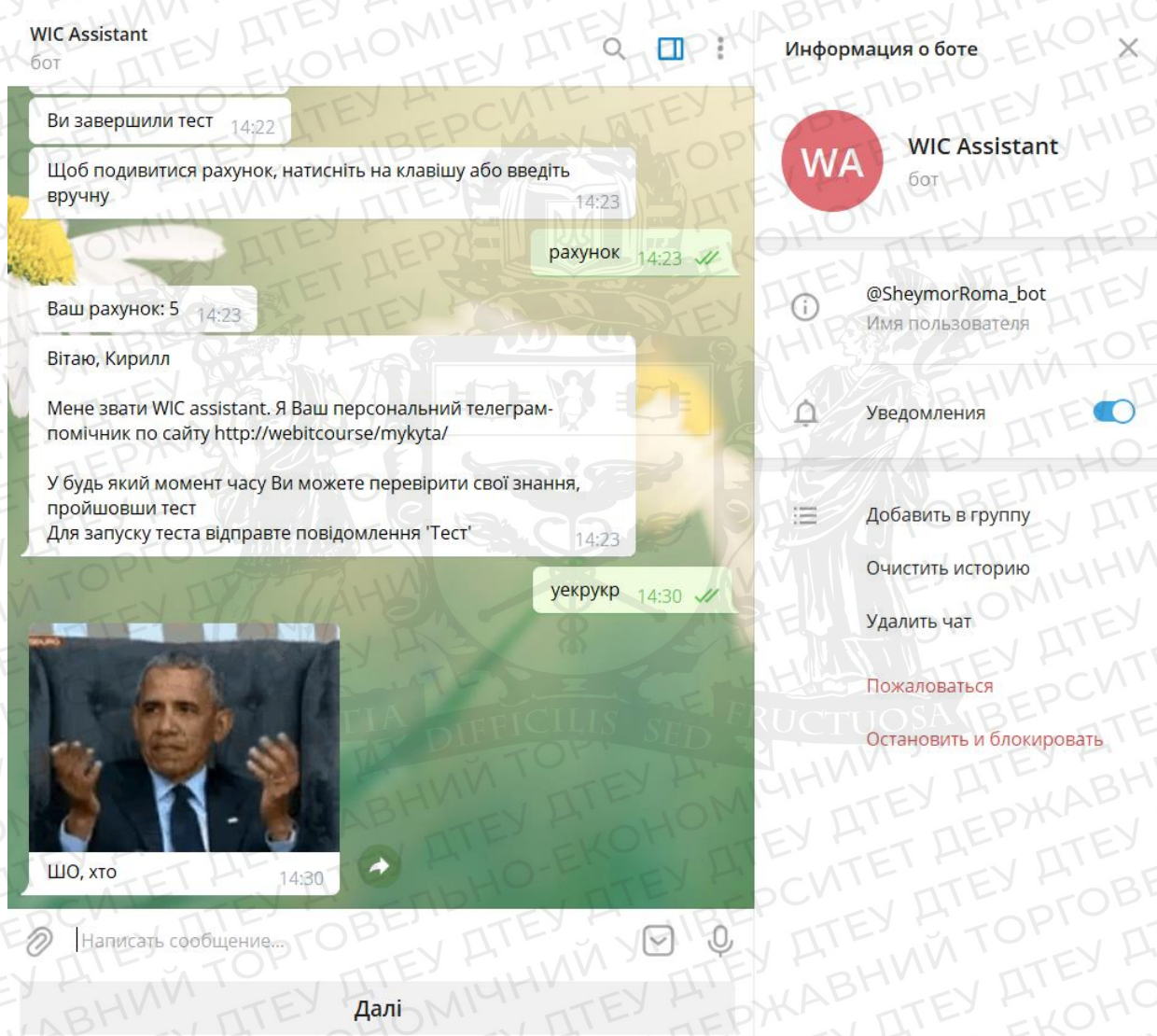


Рис 1. Введення повідомлення, яке не записане в коді

Також під час тесту заблоковані усі команди, включаючи головну команду “/start”. (рис 2).

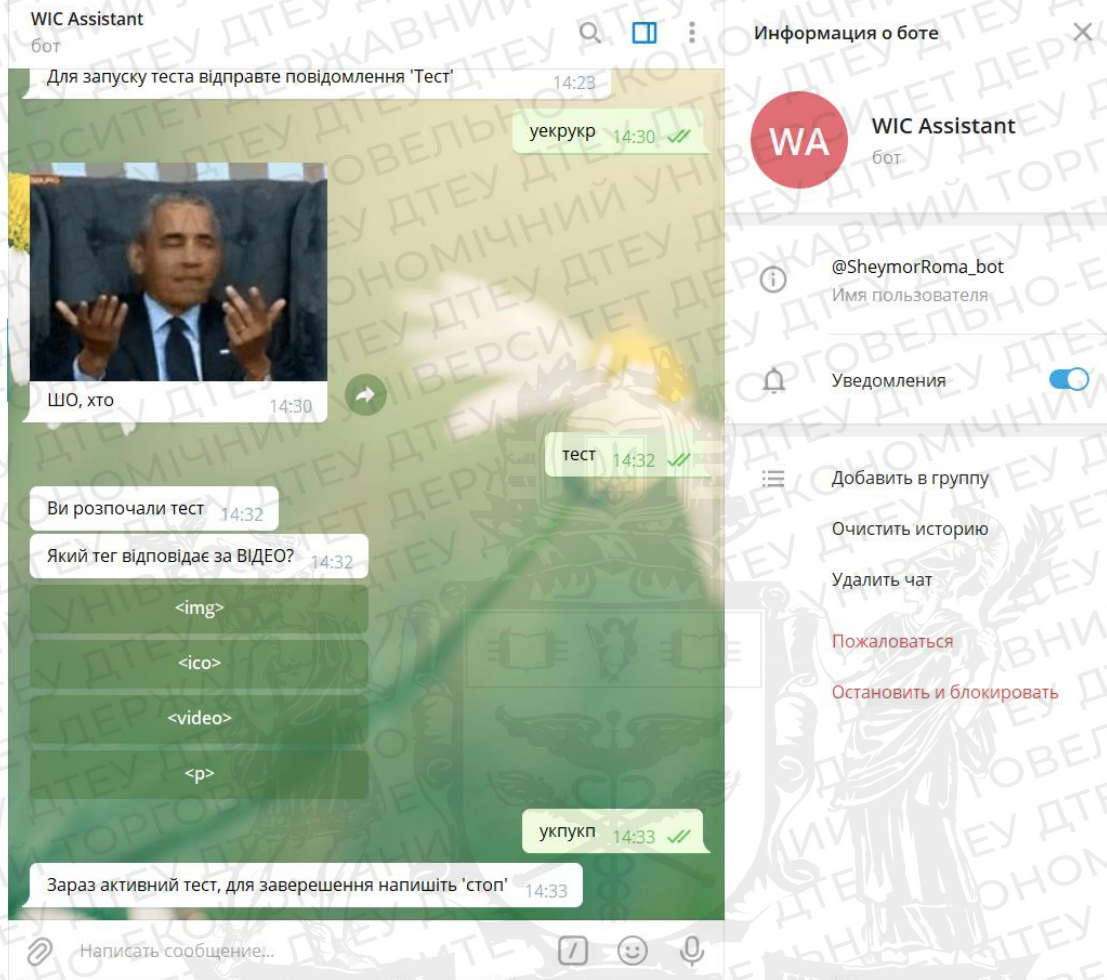


Рис 2. Введення повідомлення після початку теста

ВИСНОВКИ

Під час виконання дипломної роботи була досягнута мета: створення телеграм помічника для навчальної платформи з мов програмування. У боті було реалізовано простий діалог з посиланнями на сайт, його лекціями та тестами. Були розглянуті процес та засоби розробки телеграм помічника для навчальної платформи. Розробка цієї системи розрахована для використання лише одного сайту.

В результаті дипломної роботи були виконані наступні цілі:

1. Проведення аналізу сучасних систем розробки. На сьогоднішній день існують багато телеграм ботів, більшість з яких були створені в спеціальних конструкторах. Але зазвичай вони не є зручними, та мають свої недоліки. Такі як перевантажений функціонал, низька швидкість реагування бота, незручне та незрозуміле меню з безліччю гілок іншого меню.
2. Розглянуті суть на функції телеграм бота. У загальному представленні код розділений на 4 окремі частини. «Handlers» це головний файл, який до якого підключаються інші файли. У ньому знаходиться основний код. Наступним є файл із назвою «keyboards» у якому знаходяться усі клавіатури. Далі є файл із назвою «config» у якому прописан Token телеграм бота. Останній файл це «bot», він потрібен для збереження Token та використання його у майбутніх файлах.
3. Виконано проектування та розробку бота. Загалом, однією з важливих проблем багатьох навчальних сайтів є велика кількість інформації та посилань, у яких користувач може загубитися, для запобігання таких ситуацій і був розроблений телеграм помічник, який не тільки зменшить використання трафіку смартфона, але і полегшить “спілкування” користувача з сервісом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вікіпедія: Python (programming language) [Електронний ресурс]. - Режим доступу: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
2. Вікіпедія: General-purpose programming language [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/General-purpose_programming_language
3. Вікіпедія: High-level programming language [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/High-level_programming_language
4. Вікіпедія: Computer programming [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Computer_programming#Readability_of_source_code
5. Вікіпедія: Off-side rule [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Off-side_rule
6. Вікіпедія: Type system [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Type_system#DYNAMIC
7. Вікіпедія: ABC (programming language) [Електронний ресурс]. - Режим доступу: [https://en.wikipedia.org/wiki/ABC_\(programming_language\)](https://en.wikipedia.org/wiki/ABC_(programming_language))
8. Вікіпедія: Programming paradigm [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Programming_paradigm
9. Вікіпедія: Structured programming [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Structured_programming
10. Вікіпедія: Object-oriented programming [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Object-oriented_programming
11. Вікіпедія: Functional programming [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Functional_programming
12. Вікіпедія: Standard library [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Standard_library

13. Вікіпедія: Guido van Rossum [Електронний ресурс]. - Режим доступу:
https://en.wikipedia.org/wiki/Guido_van_Rossum
14. Вікіпедія: Garbage collection (computer science) [Електронний ресурс]. -
Режим доступу:
[https://en.wikipedia.org/wiki/Garbage_collection_\(computer_science\)](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science))
15. Вікіпедія: Backward compatibility [Електронний ресурс]. - Режим доступу:
https://en.wikipedia.org/wiki/Backward_compatibility
16. Вікіпедія: SETL [Електронний ресурс]. - Режим доступу:
<https://en.wikipedia.org/wiki/SETL>
17. Вікіпедія: Exception handling [Електронний ресурс]. - Режим доступу:
https://en.wikipedia.org/wiki/Exception_handling
18. Вікіпедія: Amoeba (operating system) [Електронний ресурс]. - Режим
доступу: [https://en.wikipedia.org/wiki/Amoeba_\(operating_system\)](https://en.wikipedia.org/wiki/Amoeba_(operating_system))
19. Вікіпедія: Cycle detection [Електронний ресурс]. - Режим доступу:
https://en.wikipedia.org/wiki/Cycle_detection
20. Вікіпедія: Metaprogramming [Електронний ресурс]. - Режим доступу:
<https://en.wikipedia.org/wiki/Metaprogramming>

ДОДАТОК

Програмний код

Heandlers

```
import logging
import time
from typing import Counter
from telebot.apihelper import convert_input_media_array
from telegram.ext.messagehandler import MessageHandler
from telegram.replymarkup import ReplyMarkup
from bot import bot
from telegram import Update
from telegram.ext import Updater, CallbackContext, dispatcher
from telegram.ext import CommandHandler
from keyboards import main_menu, reg_menu, log_menu, start_menu, choose_menu, choose2_menu, choose3_menu, end_menu, test1_menu, test2_menu, test3_menu, test4_menu, test5_menu, test6_menu, test7_menu, test8_menu, test9_menu, test10_menu, score_menu
from telebot import types

test = 0
counter = 0
sent = 0
final = 0
last = 0
last_reply = 0

@bot.message_handler(commands=['start'])
def hello(message):
    global last_reply
    global last
    if(test==0):
        print("Command start")
        last_reply = main_menu
        last = bot.send_message(chat_id=message.chat.id, text = f"Вітаю, {message.from_user.first_name}\n\n" f"Мене звати WIC assistant. Я Ваш персональний телеграм-помічник по сайту http://webitcourse/муkyta/ \n\nУ будь який момент часу Ви можете перевірити свої знання, пройшовши тест \nДля запуску теста відправте повідомлення 'Тест',reply_markup=main_menu)
    else:
        bot.send_message(chat_id=message.chat.id,text=f"Зараз активний тест, для завершення напишіть 'стоп'")
```

```

@bot.message_handler(content_types=['text'])
def bot_text(message):
    global test
    global counter
    global sent
    global last
    global last_reply
    print("Message print")
    if test==0:
        if message.text.lower() == 'дали':
            last_reply = reg_menu
            last = bot.send_message(chat_id=message.chat.id, text=f"Для початку Вам потрібно зареєструватися на сайті. Це можливо зробити перейшовши за цим посиланням \nhttp://webitcourse/муkyta/registration\_page.php",reply_markup=reg_menu)
        elif message.text.lower() == 'тест':
            print("Test start")
            bot.send_message(chat_id=message.chat.id, text=f"Ви розпочали тест",reply_markup=types.ReplyKeyboardRemove())
            sent = bot.send_message(chat_id=message.chat.id, text=f"Яким параметром ми можемо відкрити посилання в новому вікні?", reply_markup=test1_menu)
            test=1
        elif message.text.lower() == 'зареєструвався':
            last_reply = log_menu
            last = bot.send_message(chat_id=message.chat.id, text=f"Після реєстрації потрібно увійти в аккаунт сайту за посиланням \nhttp://webitcourse/муkyta/login\_page.php",reply_markup=log_menu)
        elif message.text.lower() == 'увійшов':
            last_reply = start_menu
            last = bot.send_message(chat_id=message.chat.id, text=f"Тепер, після реєстрації та входу, Вам стануть доступні лекції та тести з курсів по HTML та CSS. Готові почати?",reply_markup=start_menu)
        elif message.text.lower() == 'почати':
            last_reply = choose_menu
            last = bot.send_message(chat_id=message.chat.id, text=f"Будь ласка, оберіть з якого курсу Ви хочете почати. \nВ курсі HTML - 4 теми, в CSS - 3. \nПісля проходження всіх цих тем, стане доступна остання тема HTML/CSS",reply_markup=choose_menu)
        elif message.text.lower() == 'курс html':
            last_reply = choose2_menu
            last = bot.send_message(chat_id=message.chat.id, text=f"В курс HTML входить 4 теми: \n1. HTML основи - http://webitcourse/муkyta/lecture\_2.php \n2. Метадані HTML - http://webitcourse/муkyta/lecture\_3.php \n3. HTML РЕДАГУВАННЯ ТЕКСТУ -

```

```

http://webitcourse/mykyta/lecture_4.php \n4. HTML ГІПЕРПОСИЛАННЯ -
http://webitcourse/mykyta/lecture_5.php \n\nОберіть наступний крок",reply_markup
=choose2_menu)

    elif message.text.lower() == 'курс css':
        last_reply = choose3_menu
        last = bot.send_message(chat_id=message.chat.id, text=f"В курс CSS вх
одить 3 теми: \n1. CSS основи -
http://webitcourse/mykyta/lecture_1.php \n2. CSS ПОЗИЦІОНУВАННЯ -
http://webitcourse/mykyta/lecture_6.php \n3. CSS фон -
http://webitcourse/mykyta/lecture_7.php \n\nОберіть наступний крок",reply_markup
=choose3_menu)
    elif message.text.lower() == 'тема html/css форми':
        last_reply = end_menu
        last = bot.send_message(chat_id=message.chat.id, text=f"Тема HTML/CSS
ФОРМИ -
це остання тема, яку потрібно пройти після проходження всіх тем по HTML та CSS.
Хай щастить! \n\nТема HTML/CSS ФОРМИ -
http://webitcourse/mykyta/lecture_8.php",reply_markup=end_menu)
    elif message.text.lower() == 'закінчити':
        last_reply = start_menu
        last = bot.send_message(chat_id=message.chat.id, text=f"Вітаю, Ви зак
інчили курси по HTML та CSS! \nВи можете повернутися та пройти курси ще раз або п
іти вивчати щось нове!",reply_markup=start_menu)
    else:
        last = bot.send_video(message.chat.id, 'https://media.tenor.com/image
s/e8eaf3883926b0830870f0c89a176b55/tenor.gif', None, 'ШО, хто')
    elif test==1:
        print("Test print")
        if message.text.lower()=='стоп':
            test=0
            counter = 0
            print("Test stop")
            bot.edit_message_text(chat_id=message.chat.id, message_id=sent.mes
sage_id, text=f"Тест перервано")
            bot.send_message(chat_id=message.chat.id, text=f"Ви вийшли з тесту
")
        elif message.text.lower()=='рахунок':
            bot.send_message(chat_id=message.chat.id, text=f"Ваш рахунок: {int
(counter)}")
            if final == 1:
                test = 0
                bot.send_message(chat_id=message.chat.id, text=last.text, rep
ly_markup = last_reply)
            elif message.text.lower()=='id':

```

```

        bot.send_message(chat_id=message.chat.id, text=last_reply)
    else:
        bot.send_message(chat_id=message.chat.id, text=f"Зараз активний тест, д
ля завершення напишіть 'стоп'")

@bot.callback_query_handler (func=lambda call: call.data=='test1_1')
def test1_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Яким тегом ми можемо написати створення таблиці?',reply_markup=test2_
menu)
@bot.callback_query_handler (func=lambda call: call.data=='test1_2')
def test1_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Яким тегом ми можемо написати створення таблиці?',reply_markup=test2_
menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test1_3')
def test1_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Яким тегом ми можемо написати створення таблиці?',reply_markup=test2_
menu)
@bot.callback_query_handler (func=lambda call: call.data=='test1_4')
def test1_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Яким тегом ми можемо написати створення таблиці?',reply_markup=test2_
menu)

@bot.callback_query_handler (func=lambda call: call.data=='test2_1')
def test2_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який тег відповідає за ВІДЕО?',reply_markup=test3_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test2_2')
def test2_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який тег відповідає за ВІДЕО?',reply_markup=test3_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test2_3')
def test2_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який тег відповідає за ВІДЕО?',reply_markup=test3_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test2_4')
def test2_4(call):

```

```

        bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який теґ відповідає за ВІДЕО?',reply_markup=test3_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test3_1')
def test3_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який теґ відповідає за АУДІО?',reply_markup=test4_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test3_2')
def test3_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який теґ відповідає за АУДІО?',reply_markup=test4_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test3_3')
def test3_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який теґ відповідає за АУДІО?',reply_markup=test4_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test3_4')
def test3_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Який теґ відповідає за АУДІО?',reply_markup=test4_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test4_1')
def test4_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якими теґами виконується угрупування елементів?',reply_markup=test5_m
enu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test4_2')
def test4_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якими теґами виконується угрупування елементів?',reply_markup=test5_m
enu)
@bot.callback_query_handler (func=lambda call: call.data=='test4_3')
def test4_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якими теґами виконується угрупування елементів?',reply_markup=test5_m
enu)
@bot.callback_query_handler (func=lambda call: call.data=='test4_4')

```

```

def test4_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Якими тегами виконується угруповання елементів?',reply_markup=test5_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test5_1')
def test5_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Яка команда малює лінію в css?',reply_markup=test6_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test5_2')
def test5_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Яка команда малює лінію в css?',reply_markup=test6_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test5_3')
def test5_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Яка команда малює лінію в css?',reply_markup=test6_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test5_4')
def test5_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Яка команда малює лінію в css?',reply_markup=test6_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test6_1')
def test6_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Як задати тінь блоку в css?',reply_markup=test7_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test6_2')
def test6_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Як задати тінь блоку в css?',reply_markup=test7_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test6_3')
def test6_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Як задати тінь блоку в css?',reply_markup=test7_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test6_4')
def test6_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.id, text= f'Як задати тінь блоку в css?',reply_markup=test7_menu)

```

```

@bot.callback_query_handler (func=lambda call: call.data=='test7_1')
def test7_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якою командою в css задається позиція блоку або елемента на сторінці?'
,reply_markup=test8_menu)

    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test7_2')
def test7_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якою командою в css задається позиція блоку або елемента на сторінці?'
,reply_markup=test8_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test7_3')
def test7_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якою командою в css задається позиція блоку або елемента на сторінці?'
,reply_markup=test8_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test7_4')
def test7_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Якою командою в css задається позиція блоку або елемента на сторінці?'
,reply_markup=test8_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test8_1')
def test8_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Де слід розмішувати елемент <style>?',reply_markup=test9_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test8_2')
def test8_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Де слід розмішувати елемент <style>?',reply_markup=test9_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test8_3')
def test8_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Де слід розмішувати елемент <style>?',reply_markup=test9_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test8_4')
def test8_4(call):

```

```

bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Де слід розмішувати елемент <style>?',reply_markup=test9_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test9_1')
def test9_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Як задати розмір шрифту?',reply_markup=test10_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test9_2')
def test9_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Як задати розмір шрифту?',reply_markup=test10_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test9_3')
def test9_3(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Як задати розмір шрифту?',reply_markup=test10_menu)
    global counter
    counter=counter+1
@bot.callback_query_handler (func=lambda call: call.data=='test9_4')
def test9_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Як задати розмір шрифту?',reply_markup=test10_menu)

@bot.callback_query_handler (func=lambda call: call.data=='test10_1')
def test10_1(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Ви завершили тест')
    global final
    final = 1
    bot.send_message(chat_id=call.message.chat.id, text=f"Щоб подивитися рахунок
, натисніть на клавішу або введіть вручну",reply_markup=score_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test10_2')
def test10_2(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Ви завершили тест')
    global final
    final = 1
    bot.send_message(chat_id=call.message.chat.id, text=f"Щоб подивитися рахунок
, натисніть на клавішу або введіть вручну",reply_markup=score_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test10_3')
def test10_3(call):

```

```

bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Ви завершили тест')
global final
final = 1

bot.send_message(chat_id=call.message.chat.id, text=f"Щоб подивитися рахунок
, натисніть на клавішу або введіть вручну",reply_markup=score_menu)
@bot.callback_query_handler (func=lambda call: call.data=='test10_4')
def test10_4(call):
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.
id, text= f'Ви завершили тест')
    global counter, final
    counter=counter+1
    final = 1
    bot.send_message(chat_id=call.message.chat.id, text=f"Щоб подивитися рахунок
, натисніть на клавішу або введіть вручну",reply_markup=score_menu)

if __name__=="__main__":
    bot.polling()

```

Keyboards

```

from telebot import types

main_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True,
row_width=1)
button_1 = types.KeyboardButton(text="Далі")
main_menu.add(button_1)

reg_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True,
row_width=1)
reg_1 = types.KeyboardButton(text="Зареєструвався")
reg_menu.add(reg_1)

log_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True,
row_width=1)

```

```

log_1 = types.KeyboardButton(text="Увійшов")
log_menu.add(log_1)

start_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True, row_width=1)
start_1 = types.KeyboardButton(text="Почати")
start_menu.add(start_1)

choose_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True, row_width=1)
choose_1 = types.KeyboardButton(text="Курс HTML")
choose_2 = types.KeyboardButton(text="Курс CSS")
choose_menu.add(choose_1, choose_2)

choose2_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True, row_width=1)
choose_1 = types.KeyboardButton(text="Курс CSS")
choose_2 = types.KeyboardButton(text="Тема HTML/CSS ФОРМИ")
choose2_menu.add(choose_1, choose_2)

choose3_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True, row_width=1)
choose_1 = types.KeyboardButton(text="Курс HTML")
choose_2 = types.KeyboardButton(text="Тема HTML/CSS ФОРМИ")
choose3_menu.add(choose_1, choose_2)

end_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True, row_width=1)
end_1 = types.KeyboardButton(text="Закінчити")
end_menu.add(end_1)

test1_menu = types.InlineKeyboardMarkup(row_width=1)
test1_1 = types.InlineKeyboardButton(text="a = '_black'", callback_data='test1_1')
test1_2 = types.InlineKeyboardButton(text="target = '_black'", callback_data='test1_2')
test1_3 = types.InlineKeyboardButton(text="background = '_black'", callback_data='test1_3')
test1_4 = types.InlineKeyboardButton(text="color = '_black'", callback_data='test1_4')

test1_menu.add(test1_1, test1_2, test1_3, test1_4)

```

```

test2_menu = types.InlineKeyboardMarkup(row_width=1)
test2_1 = types.InlineKeyboardButton(text="<table> </table>", callback_data = 'test2_1')

test2_2 = types.InlineKeyboardButton(text="<tbl> </tbl>", callback_data = 'test2_2')
test2_3 = types.InlineKeyboardButton(text="<table> </tarle>", callback_data = 'test2_3')
test2_4 = types.InlineKeyboardButton(text="<b> </b>", callback_data = 'test2_4')
test2_menu.add(test2_1, test2_2, test2_3, test2_4)

test3_menu = types.InlineKeyboardMarkup(row_width=1)
test3_1 = types.InlineKeyboardButton(text="<img>", callback_data = 'test3_1')
test3_2 = types.InlineKeyboardButton(text="<ico>", callback_data = 'test3_2')
test3_3 = types.InlineKeyboardButton(text="<video>", callback_data = 'test3_3')
test3_4 = types.InlineKeyboardButton(text="<p>", callback_data = 'test3_4')
test3_menu.add(test3_1, test3_2, test3_3, test3_4)

test4_menu = types.InlineKeyboardMarkup(row_width=1)
test4_1 = types.InlineKeyboardButton(text="<audio>", callback_data = 'test4_1')
test4_2 = types.InlineKeyboardButton(text="<i>", callback_data = 'test4_2')
test4_3 = types.InlineKeyboardButton(text="<p>", callback_data = 'test4_3')
test4_4 = types.InlineKeyboardButton(text="<audia>", callback_data = 'test4_4')
test4_menu.add(test4_1, test4_2, test4_3, test4_4)

test5_menu = types.InlineKeyboardMarkup(row_width=1)
test5_1 = types.InlineKeyboardButton(text="<p> <b>", callback_data = 'test5_1')
test5_2 = types.InlineKeyboardButton(text="<img> <b>", callback_data = 'test5_2')
test5_3 = types.InlineKeyboardButton(text="<fieldset> <legend>", callback_data = 'test5_3')
test5_4 = types.InlineKeyboardButton(text="<i> <b>", callback_data = 'test5_4')
test5_menu.add(test5_1, test5_2, test5_3, test5_4)

test6_menu = types.InlineKeyboardMarkup(row_width=1)
test6_1 = types.InlineKeyboardButton(text="<hr/>", callback_data = 'test6_1')
test6_2 = types.InlineKeyboardButton(text="width=1", callback_data = 'test6_2')
test6_3 = types.InlineKeyboardButton(text="border=1", callback_data = 'test6_3')
test6_4 = types.InlineKeyboardButton(text="<tr>", callback_data = 'test6_4')
test6_menu.add(test6_1, test6_2, test6_3, test6_4)

test7_menu = types.InlineKeyboardMarkup(row_width=1)
test7_1 = types.InlineKeyboardButton(text="box-shadow", callback_data = 'test7_1')
test7_2 = types.InlineKeyboardButton(text="text-align", callback_data = 'test7_2')

```

```

test7_3 = types.InlineKeyboardButton(text="text-
decoration", callback_data='test7_3')
test7_4 = types.InlineKeyboardButton(text="text-
decoration", callback_data='test7_4')
test7_menu.add(test7_1, test7_2, test7_3, test7_4)

test8_menu = types.InlineKeyboardMarkup(row_width=1)
test8_1 = types.InlineKeyboardButton(text="text-align", callback_data='test8_1')
test8_2 = types.InlineKeyboardButton(text="position", callback_data='test8_2')
test8_3 = types.InlineKeyboardButton(text="background", callback_data='test8_3')

test8_4 = types.InlineKeyboardButton(text="color", callback_data='test8_4')
test8_menu.add(test8_1, test8_2, test8_3, test8_4)

test9_menu = types.InlineKeyboardMarkup(row_width=1)
test9_1 = types.InlineKeyboardButton(text="Всередині <link>", callback_data='tes
t9_1')
test9_2 = types.InlineKeyboardButton(text="Перед закритим тегом </body>", callbac
k_data='test9_2')
test9_3 = types.InlineKeyboardButton(text="Всередині <head>", callback_data='tes
t9_3')
test9_4 = types.InlineKeyboardButton(text="У будь-
якому місці <body>", callback_data='test9_4')
test9_menu.add(test9_1, test9_2, test9_3, test9_4)

test10_menu = types.InlineKeyboardMarkup(row_width=1)
test10_1 = types.InlineKeyboardButton(text="body { size: 15px; }", callback_data
='test10_1')
test10_2 = types.InlineKeyboardButton(text="body { fonts: 15px; }", callback_data
='test10_2')
test10_3 = types.InlineKeyboardButton(text="body { text-
size: 15px; }", callback_data='test10_3')
test10_4 = types.InlineKeyboardButton(text="body { font-
size: 15px; }", callback_data='test10_4')
test10_menu.add(test10_1, test10_2, test10_3, test10_4)

score_menu = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=Tr
ue, row_width=1)
score_1 = types.KeyboardButton(text="рахунок")
score_menu.add(score_1)

```