

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

## Пояснювальна записка

до бакалаврської кваліфікаційної роботи  
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА СНАТ-ВОТ ДЛЯ МЕРЕЖІ ШКІЛ  
ВИКЛАДАННЯ АНГЛІЙСЬКОЇ МОВИ ЗА ДОПОМОГОЮ  
PYTHON»

Виконав: студент 4 курсу, групи ПД-44

---

спеціальності 121 Інженерія програмного  
забезпечення

---

(шифр і назва спеціальності)

Шведченко І.С.

---

(прізвище та ініціали)

Керівник

Коваленко Д.С.

---

(прізвище та ініціали)

Рецензент

---

(прізвище та ініціали)

Нормоконтроль

---

(прізвище та ініціали)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного  
забезпечення

\_\_\_\_\_  
О.В. Негоденко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

**ЗАВДАННЯ**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**  
**ШВЕДЧЕНКУ ІВАНУ СЕРГІЙОВИЧУ**

1. Тема роботи «Розробка chat-bot для мережі шкіл викладання англійської мови за допомогою Python»
2. Строк подання роботи: 01.06.2021
3. Вихідні дані до роботи: якісний аналіз предметної області, проектування та розробка унікального функціонального додатка на мові програмування Python, використовуючи технологію Telegram Bot API.
4. Зміст пояснювальної записки: РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ; РОЗДІЛ 2. ПРОЕКТУВАННЯ ЧАТ-БОТА; РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТА.
5. Перелік обов'язкового ілюстративного матеріалу: презентація в *Microsoft PowerPoint*.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих прототипів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу бота	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи





## РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Чат-бот для соціальної мережі Telegram»: 49 с., 41 рис., 3 табл., 12 інформаційних джерел.

**Об'єкт розробки** – чат-бот для мережі шкіл англійської мови «Green Forest» на платформі Telegram.

**Мета проекту** – отримати готову програму – Telegram-бот для мережі шкіл англійської мови «Green Forest» для зворотного зв'язку з можливістю запису інформації, введеної користувачем, у базу даних MySQL.

**Методи розробки** – мова програмування Python, інтегроване середовище розробки PyCharm, HTTP-інтерфейс TelegramBotAPI.

СОЦІАЛЬНА МЕРЕЖА, МЕСЕНДЖЕР, РОЗРОБКА, ЧАТ-БОТ, ПОМІЧНИК, PYTHON, ПРОЕКТУВАННЯ, ЗВОТНИЙ ЗВ'ЯЗОК, ПОВІДОМЛЕННЯ, ЗАПИС НА КУРС, МЕТОД, БАЗА ДАНИХ .

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	Бази даних
URL ( <i>Uniform Resource Locator</i> )	Визначник місцезнаходження сайту в мережі Інтернет
IDE ( <i>Integrated development environment</i> )	Інтегроване середовище розробки
API ( <i>Application programming interface</i> )	Програмний інтерфейс
GFbot ( <i>Green Forest Bot</i> )	Бот для мережі шкіл англійської мови «Green Forest»
MTPROTO ( <i>Mobile Telegram Protocol</i> )	Криптографічний протокол, який використовується у Telegram для шифрування переписки користувачів
OS ( <i>Operating System</i> )	Операційна система
HTTP ( <i>HyperText Transfer Protocol</i> )	Протокол передачі даних, що використовується в комп'ютерних мережах
SQL ( <i>Structured Query Language</i> )	Мова структурованих запитів

## ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Telegram як сучасний засіб спілкування.....	9
1.2. Загальні відомості про чат-боти.....	12
1.3. Огляд існуючих чат-ботів.....	14
1.3.1. Огляд чат-боту «RailwayBot».....	14
1.3.2. Огляд чат-боту «NovaPoshtaBot».....	16
1.3.3. Огляд чат-боту «Бот «України без сміття».....	17
1.3.4. Огляд чат-боту «HotelBot».....	18
1.4. Призначення чат-бота.....	18
1.5. Опис вимог до чат-бота.....	19
2 ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ЧАТ-БОТА.....	21
2.1. Проектування функціональних можливостей чат-бота.....	21
2.2. Вибір засобів розробки.....	23
2.2.1. Мова програмування Python.....	23
2.2.2. Програмний інтерфейс Telegram Bot API.....	26
2.3. База даних MySQL.....	29
3 РОЗРОБКА ЧАТ-БОТА.....	31
3.1. Реєстрація чат-бота для Telegram Bot API.....	31
3.2. Реалізація програмної частини чат-бота.....	32
3.2.1. Створення перших повідомлення та меню.....	33
3.2.2. Розробка головного меню та його складових.....	35
3.2.3 Підключення бази даних MySQL.....	45
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А. ТЕКСТ ПРОГРАМИ.....	50
ДОДАТОК Б. ІНТЕРФЕЙС ЧАТ-БОТА.....	64



## ВСТУП

У час комп'ютерних технологій все більшої популярності набувають соціальні мережі та месенджери. Користувачі обирають зручність, доступність та простоту. Саме ці три характеристики відрізняють Telegram від конкурентів. Інформаційна система Telegram є базою такої програми, як чат-бот. Обраний напрямок є затребуваним, оскільки значну частину вільного часу люди проводять в Інтернеті, а особливо у месенджерах та соціальних мережах. І навіть обираючи між дзвінком та повідомленням, більшість обирає останнє. Саме тому бронювання квитків чи столика в ресторані, пошук фільму для перегляду, відстеження посилок – всі ці операції наразі переходять в режим онлайн, а саме в чат-боти.

Одним із найпопулярніших напрямків сьогодення, крім інформаційних технологій, є вивчення англійської мови. Україна наразі налічує нескінченну кількість шкіл та курсів, які пропонують найрізноманітніші технології та методи для засвоєння матеріалу. Майже всі школи обрали веб-сайти як платформу для розповсюдження інформації про них. Але останнім часом більшість освітніх проектів переходить у Telegram, тому що тільки таким чином освіта завжди «під рукою». Одним із таких проектів є мережа шкіл англійської мови «Green Forest». Це найсвідоміший вибір для навчання, оскільки тут поєднано якість, доступну вартість та використання комп'ютерних технологій, в тому числі і Telegram.

Метою дипломного проекту є розробка чат-бота у Telegram для мережі шкіл англійської мови «Green Forest», який забезпечуватиме зворотний зв'язок із потенційними студентами.

З одного боку, чат-бот є постійним та надійним помічником; його вимоги не виходять за рамки вимог соціальної мережі Telegram; менеджери «Green Forest» зможуть обробляти більшу кількість заявок, оскільки не будуть витрачати час, відповідаючи на загальні питання – відповідатиме чат-бот у вигляді переписки з користувачем. З іншого – потрібен тільки доступ до Інтернету, щоб користувач зміг отримати всю необхідну інформацію

незалежно від місця та часу. Тому тема розробки чат-бота є досить актуальною.

Виходячи з поставленої мети, сформовано наступні завдання:

- проаналізувати обрану предметну область;
- порівняти існуючі аналоги чат-ботів;
- обрати технології та середовище для розробки;
- розробити та протестувати чат-бота на платформі Telegram.

Проект має перспективу користуватись великим попитом у зв'язку з широкою доступністю, зручністю та зрозумілістю для пересічного користувача. У майбутньому планується вдосконалення та впровадження цілодобової роботи чат-бота за допомогою віддаленого сервера.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1. Telegram як сучасний засіб спілкування

Telegram – це кросплатформенний додаток, розроблений на мові програмування C++, який дозволяє обмінюватися повідомленнями та файлами більшості форматів. Він використовує спеціально розроблену серверну частину з закритим кодом, який працює на серверах Німеччини та США. Telegram вибивається вперед серед своїх конкурентів завдяки наступним перевагам:

- всі чати шифруються, а повідомлення знищуються через визначений термін, існує також поняття «секретний чат», повідомлення якого взагалі не зберігаються на сервері;
- повідомлення доставляються набагато швидше у порівнянні з аналогами;
- сервери Telegram розміщені по всьому світу, що значно підвищує відказостійкість;
- використання відкритого протоколу MTProto та API, безкоштовних для всіх;
- відсутність підписок та реклами;
- відсутність обмеження на розмір повідомлень та вкладених файлів.

Особливістю цієї соціальної мережі є секретні чати та спеціально розроблені протоколи шифрування MTProto (рис.1.1). Секретні чати є корисними для людей, які потребують максимально захищеного спілкування.

Окрім використання основного протоколу, всі повідомлення шифруються ключами, використовуючи принцип «від пристрою до пристрою». Ніхто не зможе перехопити та розшифрувати повідомлення, включаючи співробітників Telegram. Повідомлення з таких чатів не зберігаються на серверах додатка, їх неможливо переслати. Але завжди існує загроза витоку надсекретної інформації через скріншоти інтерфейсу Telegram з відкритим секретним чатом.

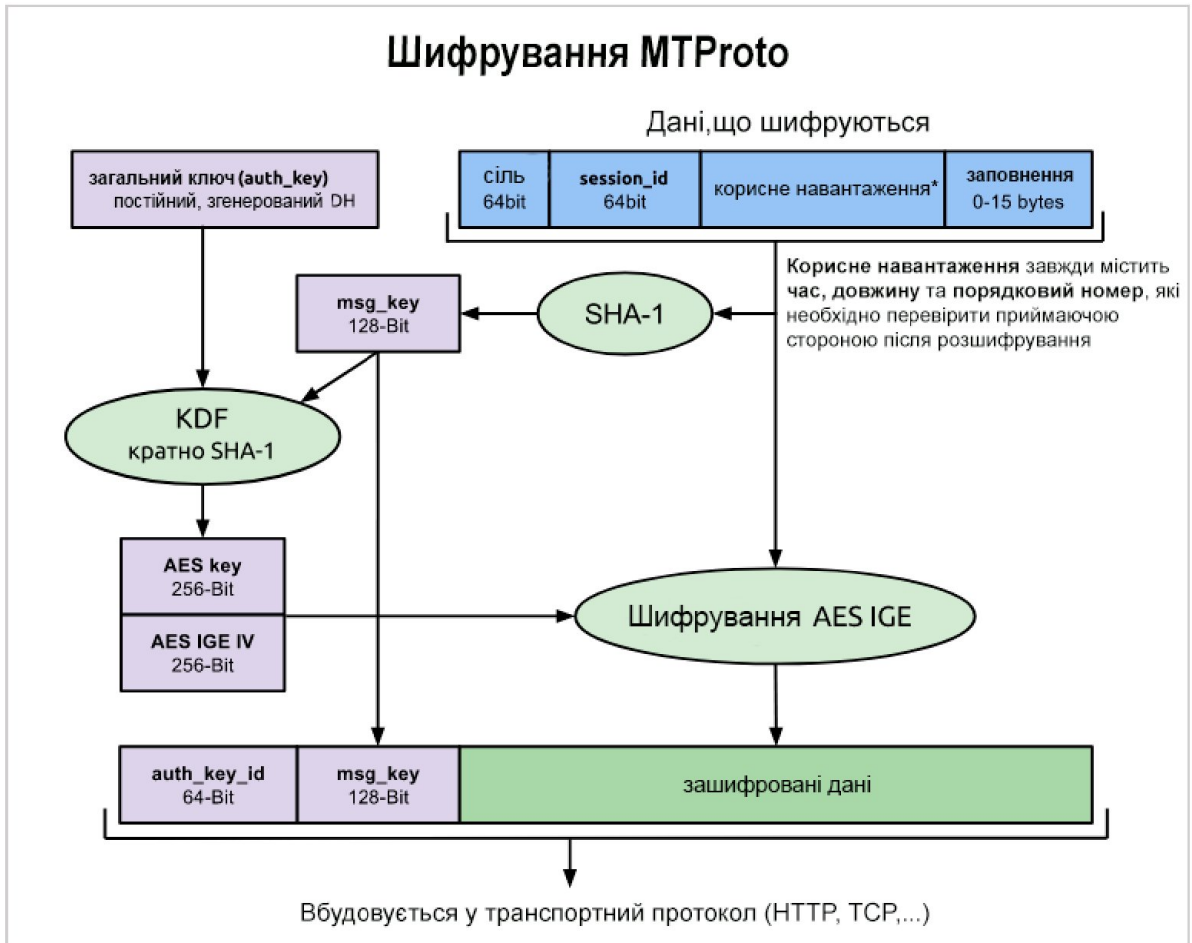


Рис.1.1. Принцип роботи протоколу шифрування MTProto

Додаток доступний для всіх популярних платформ, наприклад, Android OS, iOS, Windows, Linux тощо. У зв'язку з широкою популярністю, Telegram має в арсеналі близько 18 мов, включаючи, як мінімум, українську, англійську та німецьку (рис. 1.2 та 1.3).

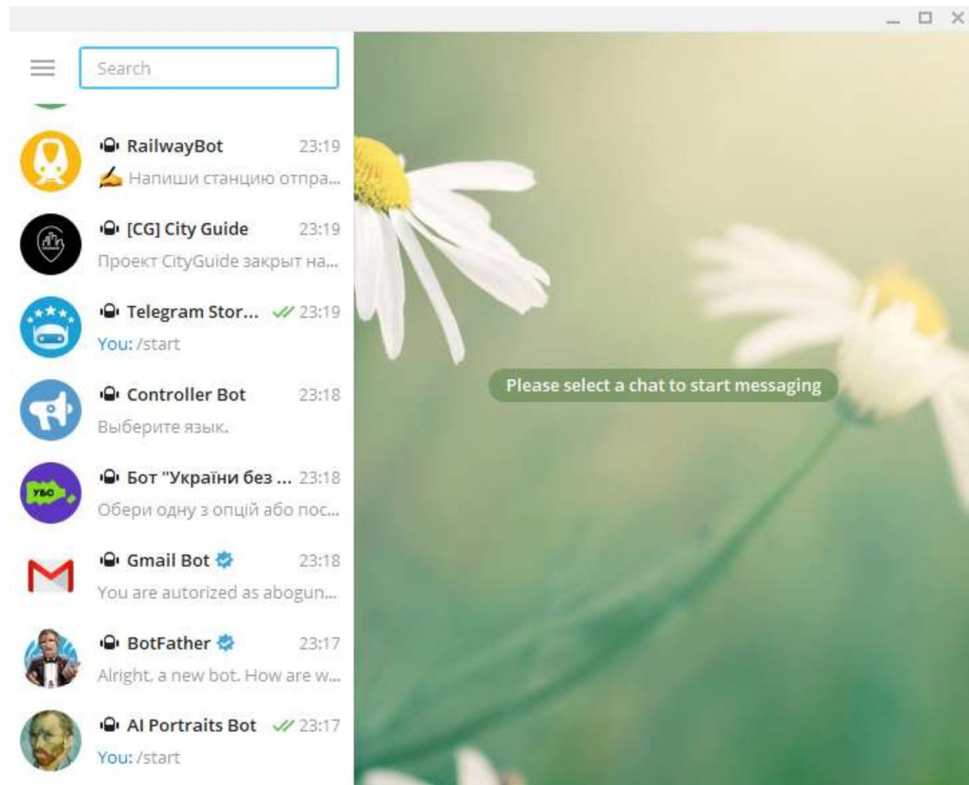
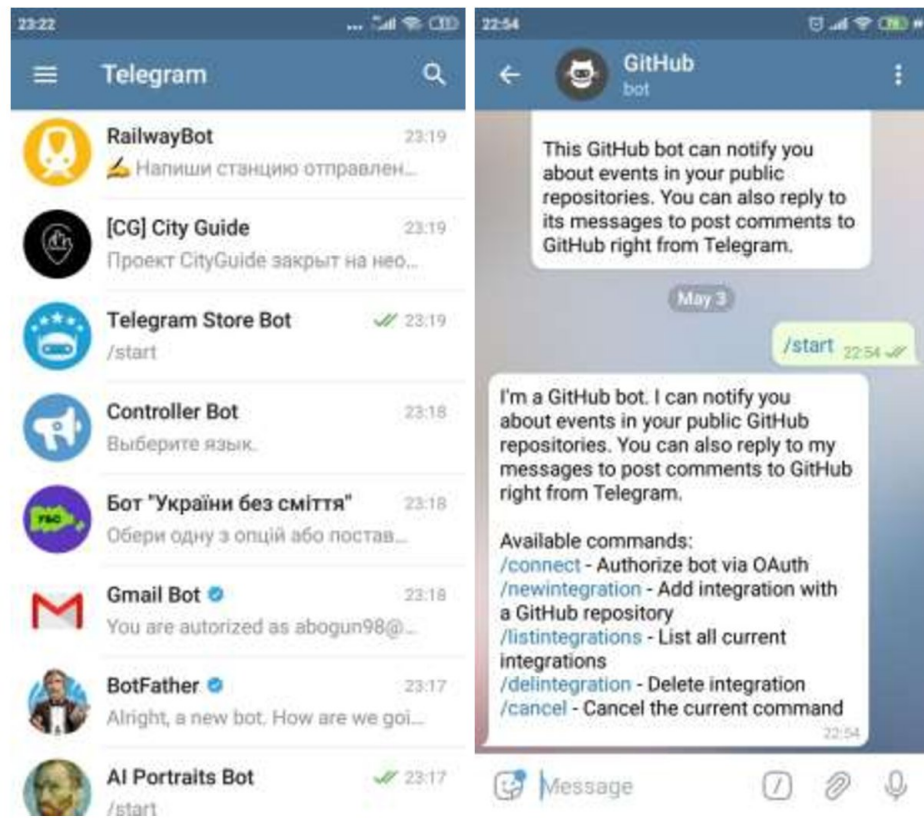


Рис.1.2. Інтерфейс Telegram на Windows 10



а)

б)

Рис.1.3. Інтерфейс Telegram на Android OS: а) список чатів; б) чат Telegram

## 1.2. Загальні відомості про чат-боти

Останнім часом в месенджерах та соціальних мережах все частіше використовуються такі сервіси як чат-боти. Дослідивши інформацію про них із декількох джерел, можна стверджувати, що чат-боти наразі є невід’ємною частиною повсякденного життя – від спілкування до розваг, від консультації лікаря до замовлення товарів та послуг, від розпізнавання емоцій до вирішення складних консалтингових задач у службах підтримки.

У будь-якому випадку, незалежно від платформи, чат-бот – це прикладна програма, яка, отримуючи інформацію від користувача, формує коректні, логічно обґрунтовані відповіді. За результатами дослідження State of Chatbots 2018 року чат-боти мають багато переваг. Основними вважаються:

- цілодобова клієнтська підтримка;
- надзвичайна швидкість та стислість відповіді;
- ефективна взаємодія з клієнтами;
- надання відповідей на прості запитання;
- уніфікований, легкий для сприйняття інтерфейс;
- можливість роботи з великою кількістю клієнтів одночасно.

Чат-боти використовуються у багатьох сферах життя.

**Бізнес.** У цій сфері чат-боти мають найбільший функціонал і можуть вирішувати багато завдань, включаючи онлайн-консультування, рекламні комунікації, продаж та ін.

**Отримання інформації.** Такі чат-боти використовуються в електронних засобах для надання інформації про, наприклад, прогноз погоди чи актуальні новини. За допомогою такого бота можна знайти заклади харчування у незнайомому місці з визначеними характеристиками.

**Навчання.** Такі боти забезпечують користувачів необхідними

навчальними матеріалами (найчастіше це невеликі за обсягом тексти) з дисципліни, оцінюють рівень знань за допомогою тестування. В даний момент чат-боти допомагають самостійно вивчати іноземні мови.

Розваги. Чат-боти цієї сфери можуть вести діалог з користувачем, грати з ним. Наприклад, перетворювати завантажені фотографії у картини.

Таким чином можна стверджувати, що роль чат-ботів з кожним днем збільшується. Це пояснюється тим, що основну частину свого часу люди проводять в смартфонах, перевіряючи пошту та повідомлення.

Боти – це програми, які виконують різноманітні завдання для користувача соціальної мережі чи месенджера. Завжди існують прості справи, на які не хочеться витратити час. Саме тут на допомогу завжди може прийти бот. Він виглядає як звичайний чат, однак спілкування відбувається не з людиною, а з програмою, яка може забронювати квитки на поїзд, якщо це бот для пошуку квитків, чи допомогти відсортувати правильно сміття, якщо це бот-сортувальник. При цьому не обов'язкового вводити дані власноруч – зазвичай діалог з ботом складається із заздалегідь зарезервованих команд, які мають вигляд кнопок (рис.1.4).



Рис.1.4. Приклад команд-кнопок для користування чат-ботом

Найбільшою перевагою чат-ботів є те, що використовувати їх надзвичайно легко. Вони створюються на базі соціальних мереж та месенджерів (Telegram, Viber, Facebook), а це означає, що вимоги чат-ботів не виходять за рамки вимог платформ, для яких вони призначені. Важливим є тільки підключення до мережі Інтернет та працюючий девайс. І навіть якщо

немає доступу до Інтернету, але переписку із ботом не видалено, користувач все ще зможе переглянути її та отримати необхідну інформацію (у випадку, якщо кеш додатка збережено).

Для того, щоб знайти потрібний чат-бот для Telegram зазвичай користуються наступними способами:

- пошук в Інтернеті;
- використовуючи пошук у Telegram та нік чат-бота.

### **1.3. Огляд існуючих чат-ботів**

Перед початком розробки чат-бота для даного дипломного проекту, ознайомимось із ринком ботів. Це потрібно для того, щоб розуміти загальну картину їх використання, уникнути помилок вже існуючих ботів та створити проект з найбільш оригінальним набором функцій. Важливо зазначити, що повних аналогів чат-бота для мережі шкіл «Green Forest» не знайдено. По-перше, це пояснюється тим, що ринок чат-ботів для обраної галузі досить новий. По-друге, не всі курси англійської потребують використання таких програм у зв'язку із невеликим потоком студентів та незначним розповсюдженням. Але оскільки бот, що розробляється, у загальному розумінні є інформаційним, розглянемо декілька популярних інформаційних чат-ботів соціальної мережі Telegram.

#### **1.3.1. Огляд чат-боту «RailwayBot»**

Чат-бот для моніторингу та пошуку квитків на українські потяги (рис. 1.5 та 1.6) є вдалим рішенням для тих, хто завжди не встигає придбати проїзні документи вчасно. Він швидко відповідає на запити, які обирає користувач: від місця до часу відправлення. У разі відсутності квитків на визначену дату, помічник пропонує створити моніторинг, який сповістить користувача про їх появу.



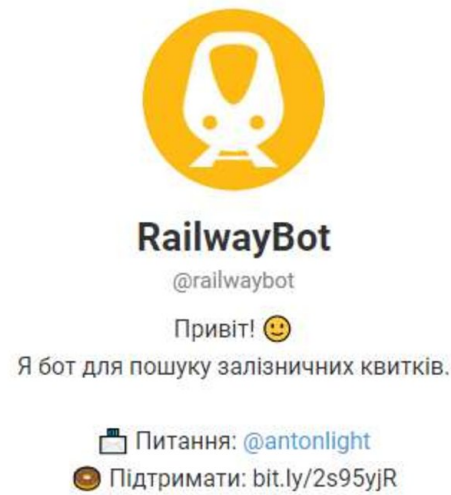


Рис.1.5. Інформація про RailwayBot

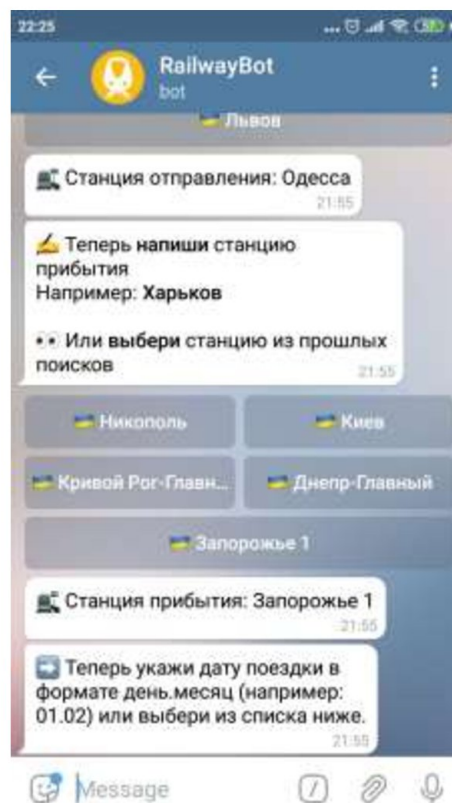


Рис.1.6. Функціонал чат-бота RailwayBot

Переваги:

- можливість моніторингу та пошуку квитків цілодобово;
- розклад потягів завжди під рукою;
- функція повернення квитків.

Недоліки:

- квитки неможливо придбати через бот;

- доступні лише російська та українська мови.

### 1.3.2. Огляд чат-боту «NovaPoshtaBot»

Це неофіційний бот «Нової Пошти», який відстежує відправлення, знаходить найближчі відділення і повідомляє про зміну статусу посилки (рис.1.7 та 1.8). Якщо поділитися з ботом номером телефону, він зможе надати більш детальний опис відправлення. Єдине, що потребує більшої уваги при вдосконаленні цього помічника – захист особистих даних користувачів.



Рис.1.7. Інформація про бот NovaPoshtaBot

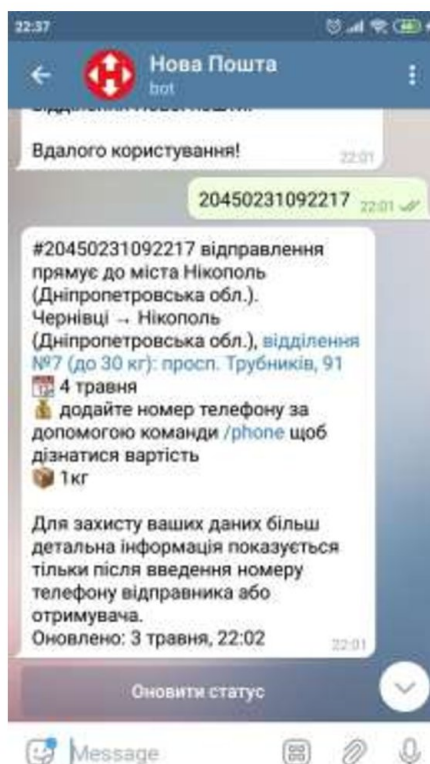


Рис.1.8. Функціонал чат-бота NovaPoshtaBot

### 1.3.3. Огляд чат-боту «Бот «України без сміття»

Бот громадської організації «Україна без сміття» (рис.1.9 та 1.10) допомагає розрізняти види вторсировини, підказує, як правильно підготувати відходи до відправки на станцію та куди їх здати. Графік прийому та адреси станцій також доступні для перегляду. «Бот вміє навчатися, тому що активніше юзер користується ним, то швидше він прогресує та відповідає на запитання», – йдеться на сторінці організації.



Рис.1.9. Інформація про бот «України без сміття»

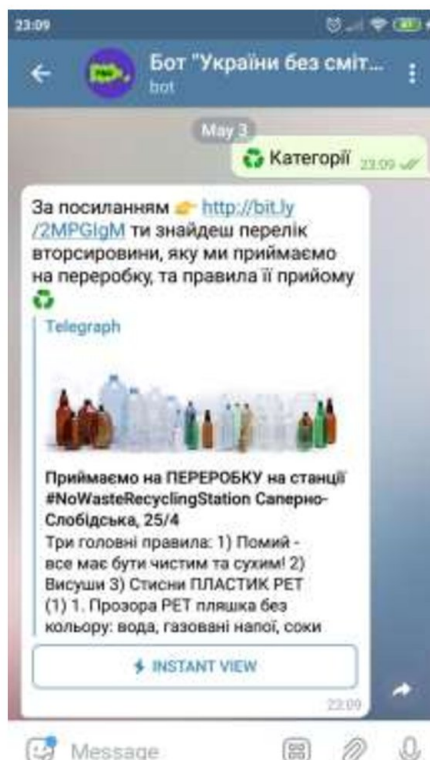


Рис.1.10. Функціонал чат-бота «України без сміття»

### 1.3.4. Огляд чат-боту «HotelBot»

Telegram-бот допомагає знайти і забронювати готельний номер. Помічник інтегрований із системою онлайн-бронювання готелів Booking.com. Після початку діалогу потрібно обрати тип пошуку готелю: за популярністю або ціною, а також вказати місто, дату заїзду, кількість ночей і осіб (рис.1.11). За цими параметрами бот шукатиме найоптимальніші варіанти. Користувачі зможуть відстежити ціну номерів у готелі: якщо вартість зміниться, вони отримають сповіщення. Бот також дає змогу переглядати фотографії помешкань, а для бронювання переадресовує користувачів на сайт Booking.com.

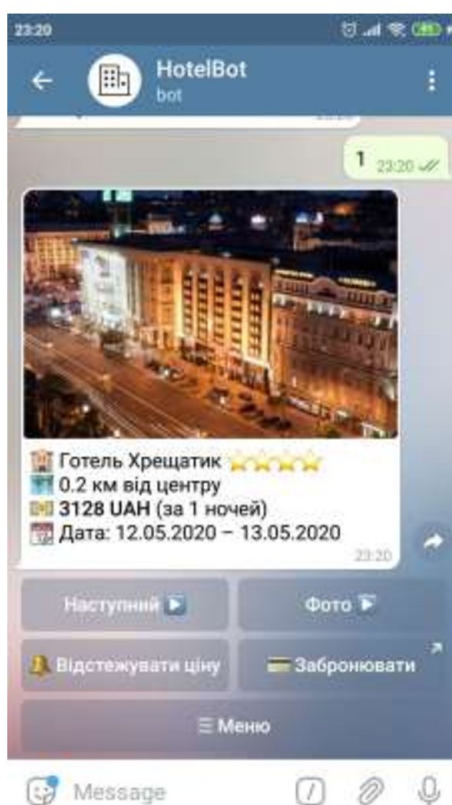


Рис.1.11. Функціонал чат-бота HotelBot

### 1.4. Призначення чат-бота

Умовним замовником чат-бота є мережа шкіл англійської мови Green Forest. Компанія є у всіх популярних соціальних мережах, в тому числі і в Telegram. Канал у Telegram використовується для публікації новин,

навчальних матеріалів, пропозицій і т. ін. Але для швидкого інформування потенційних студентів та зворотного зв'язку «Green Forest» нічого не може запропонувати. Тому прийнято рішення забезпечити популярну компанію інформаційним чат-ботом у Telegram.

Чат-бот GFbot призначений для реалізації наступних функцій:

- забезпечення інформацією потенційних студентів щодо існуючих курсів (опис курсів, ціна, розклад);
- запис на курс;
- надання інформації про місця проведення курсів з можливістю перегляду локацій на карті;
- проведення попереднього тестування для визначення рівня англійської;
- забезпечення зворотного зв'язку (функція «Замовити дзвінок»);
- надання миттєвих відповідей на найпоширеніші запитання;
- інформування студентів про проведення розмовних клубів;
- забезпечення контактами організації та шкіл в залежності від розташування.

### **1.5. Опис вимог до чат-бота**

В першу чергу, чат-бот повинен працювати у всіх версіях Telegram та на всіх пристроях, які його підтримують; повністю відповідати вимогам замовника, тобто виконувати всі перераховані раніше функції. Бот повинен мати зручний інтерфейс, мінімум тексту та максимум корисної інформації. Користувач повинен одразу зрозуміти, яким чином користуватися помічником.

Розроблений чат-бот повинен забезпечувати надійність, захищеність та цілісність даних, які вводяться користувачем. Інформація, представлена у помічнику, повинна відповідати достовірності та актуальності.

У разі виникнення помилок чи будь-якої небажаної поведінки чат-бота

користувач повинен мати змогу зв'язатися з адміністратором помічника. Будь-який девайс зі встановленим додатком Telegram та доступ до Інтернету – єдині вимоги, які потрібно виконати для користування чат-бот

## 2 ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ЧАТ-БОТА

### 2.1. Проектування функціональних можливостей чат-бота

Чат-бот, що розроблювався у дипломному проєкті, має наступні функціональні характеристики:

- навігація за допомогою кнопок та головного меню;
- надання потрібної інформації користувачеві при використанні запитів;
- зручний та візуально приємний інтерфейс;
- робота на будь-якому пристрої, який підтримує Telegram з підключенням до Інтернету;
- швидка відповідь на запит користувача.

Робота чат-бота починається з команди /start. Перше, що бачить користувач після запуску бота – запит про вибір міста. Після цього стає доступним головне меню.

Розглянемо більш детально можливості та функції чат-бота в залежності від розділів головного меню.

*«Обрати курс»*. Вибір курсу, перегляд його опису (час проведення, вартість, переваги). Користувач може вибрати всі значення і тільки після перегляду всіх обрати бажаний курс. Під інформацією про можливі курси знаходяться кнопки «Записатися на курс» та «Записатися на пробне заняття».

Якщо користувач обрав перше значення, бот просить послідовно ввести необхідну для реєстрації інформацію, а саме: обрати бажані школу та час, ввести ПІБ, номер телефону, рівень англійської (за необхідності пройти запропонований тест). Якщо друге – виконати ті ж самі команди, крім введення часу.

Під час вибору школи та після запису на курс/пробне заняття користувач може переглянути, де знаходиться школа на карті.

*«Розклад розмовних клубів».* Перелік можливих розмовних клубів від «Green Forest» та умови їх відвідування. Бонусом є посилання на канал, на якому публікується інформація про розмовні клуби в залежності від міста.

*«Найпоширеніші запитання».* Перелік популярних запитань зі змістовними короткими відповідями. За допомогою цієї функції менеджери школи не витратять час, відповідаючи на прості загальні запитання.

*«Замовити дзвінок».* Функція, яка дозволяє залишити свій номер телефону та коментар (зручний час для дзвінка, питання тощо) для того, щоб менеджер школи зв'язався з користувачем.

*«Контакти».* Перелік усіх існуючих контактів компанії (від телефонних номерів до посилань на соціальні мережі).

*«Повернутися до вибору міста».* Функція, яка дозволяє змінити місто користувача за необхідності.

Закінчити роботу з помічником можливо шляхом припинення користування його командами (кнопками). Для того, щоб повністю видалити чат-бот та діалог з ним потрібно скористуватися можливостями Telegram.

На рисунку 2.1 відображено загальну діаграму варіантів використання (прецедентів), на якій зображено відносини між акторами (користувачем, чат-ботом та базою даних) та варіантами використання (прецедентами – функціональними можливостями акторів).





Рис.2.1. Загальна діаграма варіантів використання

## 2.2. Вибір засобів розробки

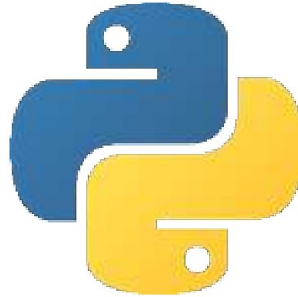
### 2.2.1. Мова програмування Python

Одним з найважливіших завдань під час створення дипломного проекту є правильний вибір засобів розробки. В результаті аналізу мов програмування, які зазвичай використовуються для створення чат-ботів, визначено, що Python є найбільш вдалим варіантом для даного проекту.

Python – інтерпретована, високорівнева, інтерактивна, об'єктно-орієнтована скриптова мова програмування (рис.2.2). Вона розроблена як мова, що «легко читається»; часто використовує в якості ключових слів слова англійської мови.

Вихідний код на Python НЕ компілюється в машинний код, а виконується безпосередньо за допомогою спеціальної програми-

інтерпретатора. Користувач може писати код прямо в оболонці інтерпретатора і вводити нові команди у міру того, як виконуються попередні. Інтерпретатор починає свою роботу у верхній частині файлу, перекладає перший рядок на машинну мову, а потім виконує його. Цей процес повторюється до кінця файлу. На рисунку 2.3 зображено як працює інтерпретатор.



2.2. Логотип мови програмування Python

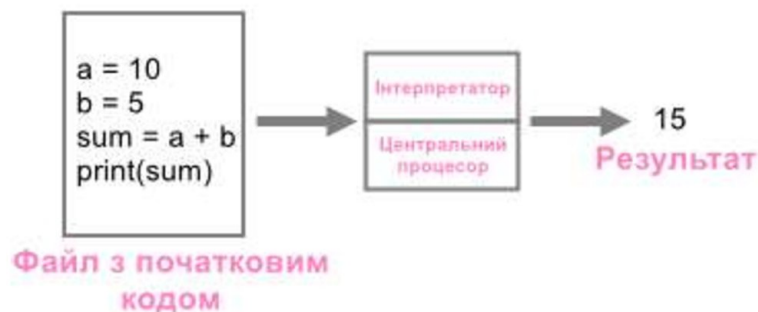


Рис.2.3. Виконання програми інтерпретатором

Розробка мови Python припадає на кінець 80-их років співробітником Голландського Національного Дослідницького Інституту Математики та Інформатики Гвідо ван Россумом.

Мова програмування Python увібрала у себе риси багатьох популярних в той час мов: Algol-68, C, C ++, Modula-3 ABC, SmallTalk, і це далеко не повний перелік. Python підтримує принципи ООП, а саме інкапсуляцію коду в особливі структури, іменовані об'єктами.

Версія 1.0 з'явилася в 1994 році, 2.0 в 2000-му, а 3.0 в 2008-му році. Підтримка Python'a здійснюється командою розробників все того ж інституту, при цьому за ван Россумом залишилося право вирішального голосу в питаннях

розвитку мови.

При розробці дипломного проекту використовувалася найсучасніша версія – Python 3.8.2 2020 року.

### ***Особливості мови Python:***

- *легка для вивчення:* у мови мало ключових слів, проста структура і чітко визначений синтаксис, завдяки цьому навчитися основам Python'a можливо за досить короткий час;
- *легко читається:* блоки коду в Python виділяються за допомогою відступів, що спільно із ключовими словами, взятими з англійської мови, значно полегшують читання коду;
- *легка в обслуговуванні:* однією з причин широкої популярності Python'a є простота обслуговування коду, написаного на цій мові;
- *широка стандартна бібліотека:* наявність широкої крос-платформної бібліотеки є ще однією сильною стороною цієї мови програмування;
- *наявність інтерактивного режиму:* дозволяє швидко тестувати потрібні ділянки коду;
- *портативність:* Python без проблем запускається на різних платформах, при цьому зберігає однаковий інтерфейс, незалежно від того, на якому комп'ютері працює користувач;
- *можливість розширення:* за необхідності в Python можна впроваджувати низькорівневі модулі, написані на інших мовах програмування, для найбільш гнучкого вирішення поставлених завдань;
- *робота з базами даних:* в стандартній бібліотеці Python можна знайти модулі для роботи з більшістю комерційних баз даних;
- *створення GUI (графічного інтерфейсу користувача):* на Python можливе створення GUI додатків, які працюватимуть незалежно від типу операційної системи.

### 2.2.2. Програмний інтерфейс Telegram Bot API

Telegram Bot API - це HTTP-інтерфейс для роботи з ботами у Telegram. По суті бот – це спеціальний обліковий запис, створений для автоматичної відправки та обробки повідомлень.

Для роботи з Telegram Bot API вивчено документацію, в якій описані всі методи і параметри, що передаються. Існує два протилежних за логікою способи отримання оновлень від бота.

1) `getUpdates`(використовуючи `long polling`) використовується для отримання оновлень через `long polling`. Логіка полягає у періодичному опитуванні серверів Telegram на предмет наявності нової інформації. З'єднання відкривається на нетривалий час і всі оновлення одразу відправляються боту. Відповідь повертається у вигляді масиву об'єктів `Update`. Параметри методу наведені у таблиці 2.1.

Табл.2.1.

Параметри методу `getUpdates`

Параметри	Тип	Обов'язковий	Опис
<code>offset</code>	Integer	Ні	Дозволяє переглядати нові повідомлення, ігноруючи старі.
<code>limit</code>	Integer	Ні	Обмежує кількість елементів масиву <code>Update</code> . Приймає значення від 1 до 100.
<code>timeout</code>	Integer	Ні	Таймаут в секундах для <code>long polling</code> . За замовчуванням 0, тобто короткий запит.

2) `setWebhook` необхідний для задання URL вебхука, на який бот відправлятиме оновлення. Мається на увазі, що тепер якщо в чат приходять

повідомлення, то Telegram сам говорить про це. Кожний раз при отриманні оновлення на цю адресу відправлятиметься HTTPS POST із JSON-об'єктом Update. При невдалому запиті до сервера спроба повториться декілька разів. Для більшої безпеки рекомендується включити токен в URL вебхука, наприклад, наступним чином: `https://yourwebhookserver.com/<token>`. Для того щоб отримати токен (власний ключ авторизації) необхідно написати спеціальному боту @BotFather. Оскільки ніхто сторонній не знає особистий токен користувача, запити до вебхуку посилає саме Telegram. Параметри методу наведені в таблиці 2.2.

Табл.2.2.

Параметри методу setWebhook

Параметри	Тип	Обов'язковий	Опис
url	String	Ні	HTTP url для відправки запитів. Щоб видалили вебхук, треба відправити порожній рядок.
certificate	InputFile	Ні	Завантаження публічного ключа для перевірки кореневого сертифікату.

*Примітка:*

- при підключеному та налаштованому вебхуці метод getUpdates не працюватиме;
- при використанні підписаного власноруч сертифікату, вам необхідно завантажити публічний ключ за допомогою параметра certificate;
- на сьогоднішній день відправка оновлень через вебхуки доступна тільки на ці порти: 443, 80, 88, 8443.

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS в

наступному вигляді: [https://api.telegram.org/bot<токен>/НАЗВА\\_МЕТОДУ](https://api.telegram.org/bot<токен>/НАЗВА_МЕТОДУ).

Принцип взаємодії чат-бота та користувача зображено на рисунку 2.4.

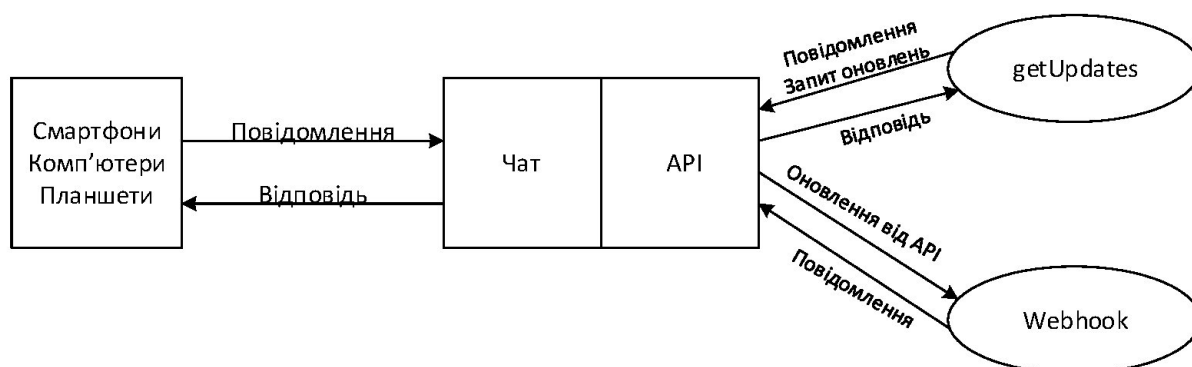


Рис.2.4.Принцип роботи чат-бота на платформі Telegram

Приклади інших доступних для Telegram Bot API методів описані нижче:

- метод для відправки текстових повідомленнях *sendMessage*;
- метод для відправки точки на карті *sendLocation*;
- метод для редагування текстових повідомлень, відправлених ботом чи за допомогою бота *editMessageText*;
- метод для відправки аудіофайлів, якщо бажаєте, щоб клієнти Telegram відображали їх в музичному плеєрі *sendAudio*;
- метод для відправки телефонних контактів *sendContact*.

Допускаються POST і GET запити. Існує 4 способи для передачі параметрів в Bot API:

- запит в URL;
- application / x-www-form-urlencoded;
- application / json (не прийнятний для завантаження файлів);
- multipart / form-data (для завантаження файлів).

### 2.3. База даних MySQL

Для повного виконання вимог замовника, чат-бот повинен не тільки володіти інформацією та показувати її користувачеві, а ще й записувати та зберігати її. У даному дипломному проєкті ключовим є запис потенційних студентів на курс, пробне заняття чи просто для консультації та зберігання цих даних для подальшої обробки інформації менеджерами. Тут і виникає потреба у тому, щоб вибрати найбільш зручний та сучасний спосіб для зберігання великого обсягу особистої інформації. В результаті дослідження цього питання, обрано СУБД компанії Oracle – MySQL. Але перед початком роботи, ознайомимось з фундаментальними поняттями.

База даних – це централізоване сховище даних, яке забезпечує зберігання, доступ, первинну обробку та пошук інформації. Це упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів.

База даних повинна забезпечувати:

- незалежність даних;
- відкритий доступ до даних;
- підтримку транзакцій з гарантією відповідних властивостей;
- гарантовану відсутність збоїв;
- одночасну роботу з багатьма користувачами.

MySQL – це система управління реляційними базами даних з відкритим вихідним кодом. Має клієнт-серверну архітектуру: до сервера MySQL можуть звертатися різні клієнтські додатки, у тому числі з віддалених комп'ютерів.

Найважливіші особливості, завдяки яким ця програма набула популярності, це:

- відкритість коду: будь-хто може безкоштовно завантажити програму на сайті розробника та при необхідності допрацювати її;
- кросплатформність: її можна використовувати майже у всіх операційних системах;

- наявність програмних інтерфейсів (API): до БД можуть підключатися додатки, створені за допомогою C/C++, EUFFEL, JAVA, Visual Studio, Python та ін.;
- чудові технічні характеристики: багатопоточність, багатокористувацький доступ, швидкодія, масштабованість ( наприклад робота з 60 тис. таблиць, які містять приблизно 5 млрд рядків);
- розвинена система безпеки та розмежовування доступу на основі системи привілеїв.



## 3 РОЗРОБКА ЧАТ-БОТА

### 3.1. Реєстрація чат-бота для Telegram Bot API

Після проектування чат-бота переходимо до розробки описаними у попередньому розділі засобами.

Перший крок – реєстрація чат-бота, що розробляється, у Telegram за допомогою головного бота «BotFather». Він призначений для створення нових ботів та керування ними. Його дуже просто знайти, використовуючи пошук Telegram. Після початку роботи з ним за допомогою команди «/start» зі списку команд потрібно обрати «/newbot», що передбачає створення власного бота. Після цього варто ввести назву, яка міститиме в собі слова «Bot» чи «\_bot». Якщо назву введено правильно, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) та URL-адресу для доступу до чат-бота. Процес реєстрації GFbot проілюстровано на рисунку 3.1.

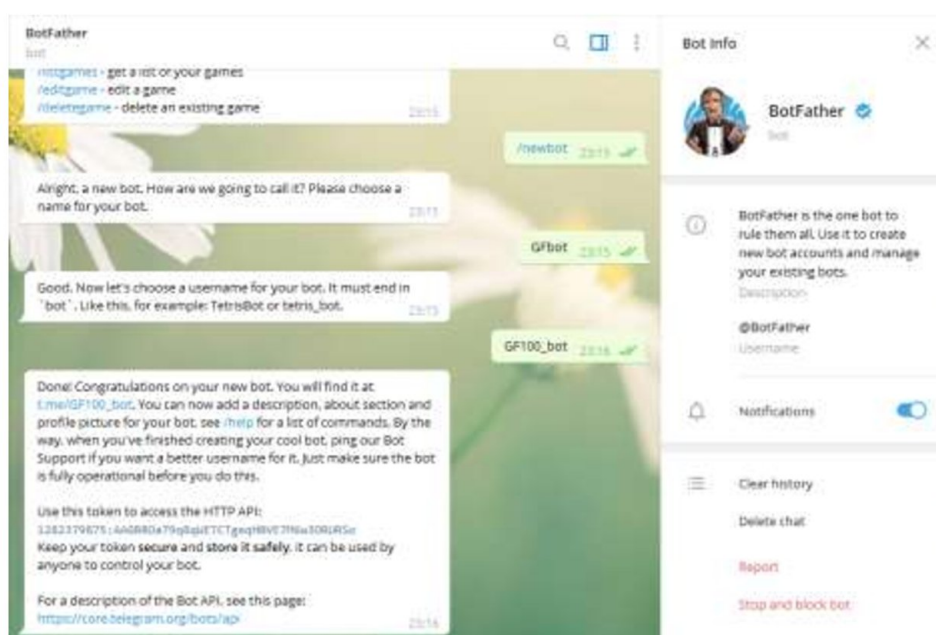


Рис.3.1. Реєстрація GFbot у Telegram

Для редагування параметрів чат-бота, наприклад, зображення або опису, існує ряд команд (таблиця 3.1). Інтерфейс GFbot приведено в додатку Б.

Табл.3.1.

#### Доступні команди для редагування чат-ботів

Команда	Опис
/setname	Змінює існуюче ім'я
/setdescription	Встановлює текст, який відображається при першому відкритті чат-бота
/setabouttext	Встановлює текст у поле «Про чат-бот»
/setuserpic	Встановлює обране зображення
/setcommands	Дозволяє створити список доступних команд
/deletebot	Видаляє обраного чат-бота

Після налаштування бота зі сторони Telegram та отримання токена можна переходити до розробки програмної частини.

### 3.2. Реалізація програмної частини чат-бота

Проект складається з файлів, зображених на рисунку 3.2, які утворюють програмну реалізацію чат-бота на мові програмування Python. Для отримання оновлень обрано спосіб з Long Polling, оскільки його реалізація достатньо проста та не потребує особливих умов (бот можна запустити на будь-якому комп'ютері, зворотна адреса не потрібна).

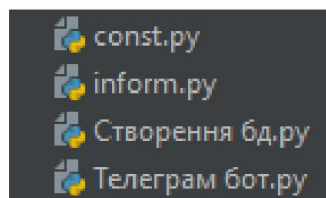


Рис.3.2. Файлова структура чат-бота у PyCharm

Розглянемо кожний файл більш детально:

- const.py : містить константи, які використовуються у проекті;
- inform.py : вміщує всю інформацію, яка необхідна для роботи чат-бота, починаючи від токена та закінчуючи графіком роботи шкіл «Green Forest»;
- файл Створення бд.py : підключення до серверу MySQL, створення та заповнення бази даних після роботи з відповідними командами чат-бота;
- файл Телеграм бот.py ; основний файл, у якому відбувається створення основних візуальних складових чат-бота та прописується логіка його роботи.

Перед початком створення скрипту для бота, необхідно завантажити бібліотеку pyTelegramBotAPI та підключити її до проекту з іншими модулями, які потрібні для розробки(рис.3.3).

```
import mysql.connector
import telebot
from telebot import types
import datetime
import const
from inform import token, hello, save_answer, choose_city,
```

Рис.3.3. Підключення бібліотек та необхідних модулів для роботи

### 3.2.1. Створення перших повідомлення та меню

Для початку роботи чат-бота потрібно відправити йому команду «/start» (або «/help»), яка є єдиною для користувача після вибору діалогу з помічником. Але для того, щоб бот сприймав цю команду та починав працювати після неї, створено декоратор.

Декоратор – це функція, яка дозволяє змінити функціональність іншої функції у деякому напрямку, при цьому не змінюючи її код. Приклад декоратора можна побачити на рисунку 3.4. Він виступає в ролі «слухача», який обробляє вхідні повідомлення.

```
# Початок роботи бота /start or /help. Вибір міста
@bot.message_handler(commands=['help', 'start'])
```

Рис.3.4. Використання декоратора для початку роботи бота

Після обробки цієї команди, бот запропонує користувачеві вибрати місто, щоб інформація в усіх інших розділах відповідала обраному значенню . За допомогою методу *send\_message* помічник відправить у чат перше повідомлення та запропонує обрати місто зі створеного списку-клавіатури. Параметри методу для створення клавіатури в цілому *ReplyKeyboardMarkup* вказують на ширину рядків (*row\_width*) та на те, що клавіатура змінюватиме свій розмір в залежності від кількості кнопок для кращого її сприйняття користувачем (*resize\_keyboard=True*). Для скривання клавіатури використовується метод *ReplyKeyboardRemove*.

На рисунку 3.5 зображено програмний код, який описує процес відправки першого повідомлення чат ботом та створення першої клавіатури, що виступає в ролі меню з кнопками, а на рисунку 3.6 – результат роботи цього фрагменту. Після вибору міста значення зберігається у змінну.

```
markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
itembtn1 = types.KeyboardButton(city_1)
itembtn2 = types.KeyboardButton(city_2)
itembtn3 = types.KeyboardButton(city_3)
itembtn4 = types.KeyboardButton(city_4)
itembtn5 = types.KeyboardButton(city_5)
markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5)
bot.send_message(message.chat.id, hello, reply_markup=markup)
```

Рис.3.5. Створення першого повідомлення та списку міст

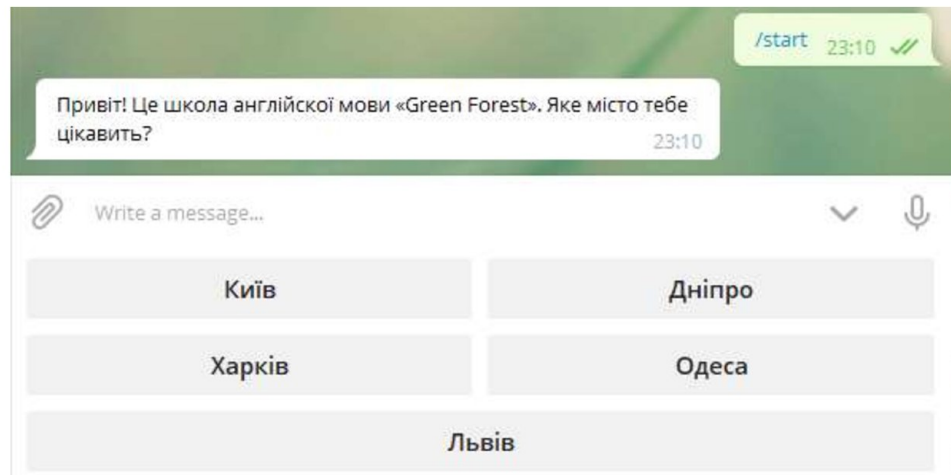


Рис.3.6. Вибір міста після початку роботи бота

### 3.2.2. Розробка головного меню та його складових

Головне меню містить:

- розділ «Обрати курс»;
- розділ «Розклад розмовних клубів»;
- розділ «Найпоширеніші запитання»;
- розділ «Замовити дзвінок»;
- розділ «Контакти»;
- розділ «Повернутися до вибору міста».

Створюється аналогічно до меню з містами, а з'являється після того, як користувач вибрав місто та бот його записав у змінну. Після отримання відповіді від користувача щодо міста, помічник відповідає «Дякую, я запам'ятав! Обери, яке питання тебе цікавить нижче.» та дає змогу побачити головне меню (рис.3.7).

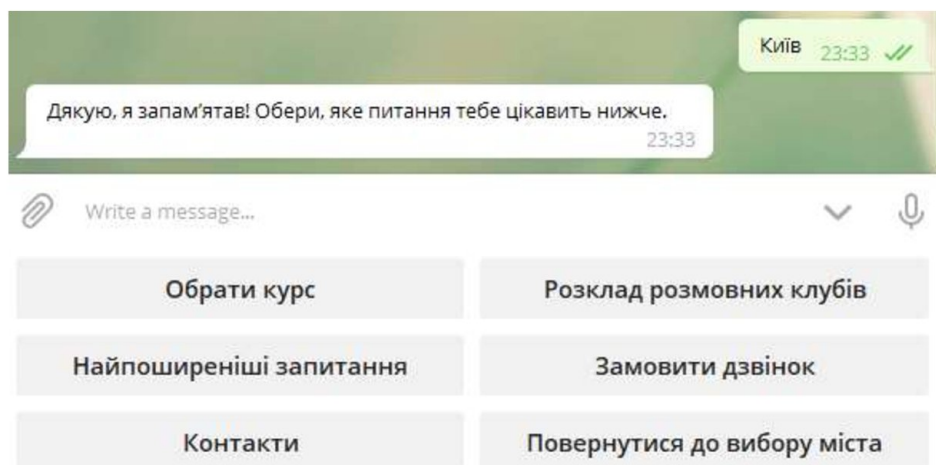


Рис.3.7. Головне меню

### *Розділ «Обрати курс»*

Даний розділ є основним, оскільки він виконує головне завдання – реєстрацію на курс. Складається з наступних пунктів:

- інтенсив вихідного дня;
- стандартний курс;
- інтенсивний курс;
- на головну.

Має вигляд меню, створеного аналогічно до першого меню з вибором міста (рис.3.8). Пропонує користувачеві обрати для себе курс. Вибір певного курсу можливий після перегляду всіх запропонованих, тому користувача нічого не зобов'язує просто переглянути інформацію про послуги, які пропонує школа «Green Forest».

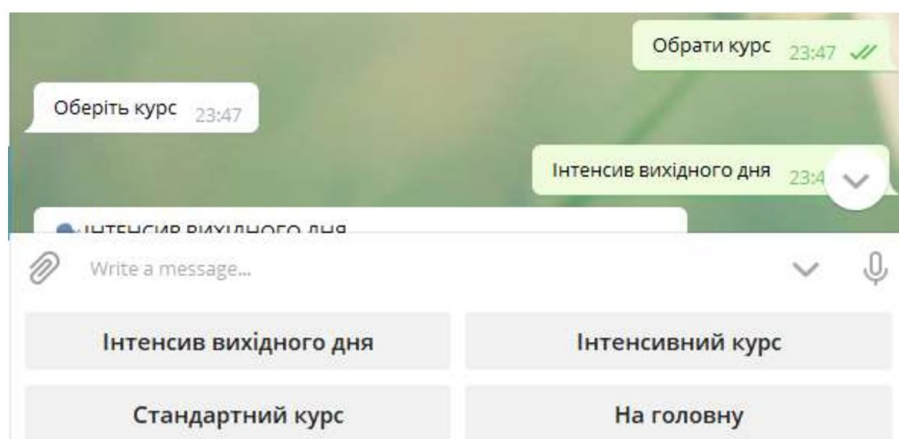


Рис.3.8. Складові розділу «Обрати курс»

Розглянемо на прикладі стандартного курсу логіку програми та роботу чат-бота.

Як тільки користувач вибирає для себе курс, який він хоче переглянути, GFbot миттєво відповідає йому повідомленням у чаті з потрібною інформацією (рис.3.9). Меню розділу «Обрати курс» не зникає.

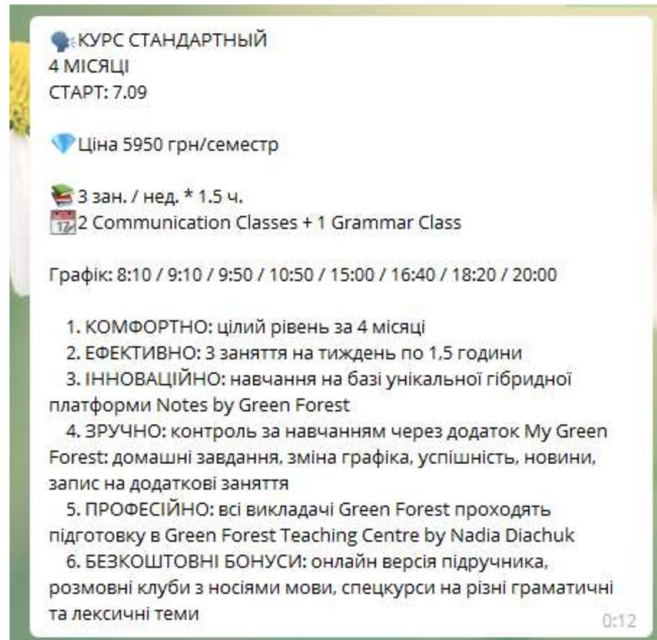


Рис.3.9. Інформація про стандартний курс

Всі дані знаходяться у файлі *inform.py* та прив'язані до певних змінних, що дозволяє виводити будь-який обсяг інформації при цьому не навантажуючи код простим текстом. У мові Python не передбачений вибір типу тексту (курсив, жирний), але є можливість використовувати смайли та пробіли (рис 3.10).

```

# Курс СТАНДАРТИЙ
Standard = ''
📌 КУРС СТАНДАРТНЫЙ
4 МІСЯЦІ
СТАРТ: 7.09

📌 Ціна 5950 грн/семестр

📌 3 зан. / нед. * 1.5 ч.
📌 2 Communication Classes + 1 Grammar Class

Графік: 8:10 / 9:10 / 9:50 / 10:50 / 15:00 / 16:40 / 18:20 / 20:00

```

Рис.3.10. Вигляд тексту зі смайлами у програмному коді

Для переходу до наступного кроку після отримання інформації, за допомогою методу *InlineKeyboardMarkup* створюється нова клавіатура – вбудована. Вона знаходиться не у нижній частині діалогу (як головне меню, наприклад), а в чаті під текстом, до якого належить. Відповідно за допомогою методу *InlineKeyboardButton* створюються кнопки цієї клавіатури. Фрагмент коду, у якому створюється вбудована клавіатура зображений на рисунку 3.11. Як виглядає вбудована клавіатура у Telegram можна побачити на рисунку 3.12.

```

# Курс Стандартний. Створення кнопок: Записатися на курс, Записатися на пробне заняття
@bot.message_handler(func=lambda mess: mess.text=='Стандартний курс' and mess.content_type=='text')
def Standard_course(message):
    bot.send_message(message.chat.id, Standard)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=enroll_course, callback_data='button5_1')
    button_2 = types.InlineKeyboardButton(text=enroll_trial_lesson, callback_data='button6_1')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(message.chat.id, make_choice, reply_markup=keyboard)

```

Рис.3.11. Створення вбудованої клавіатури



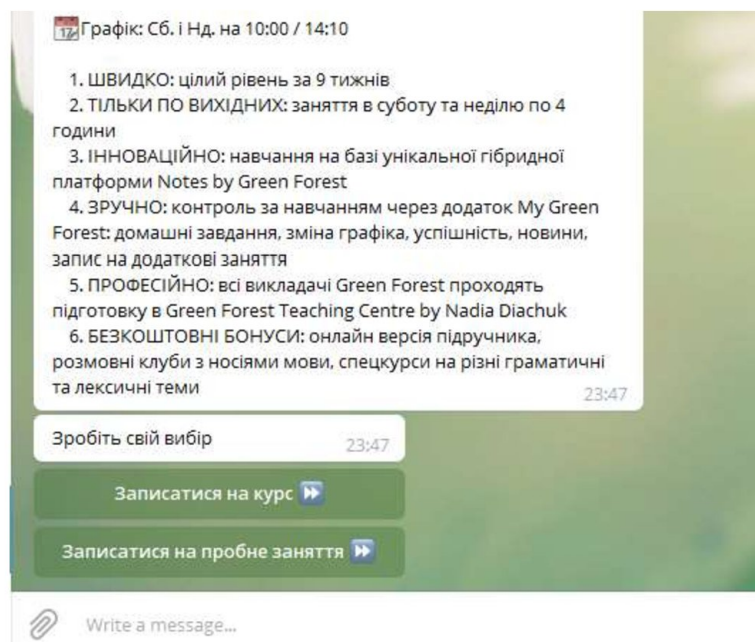


Рис.3.12. Реалізація вбудованої клавіатури у Telegram

Для продовження роботи з ботом, а саме щоб записатися на курс чи пробне заняття, створено обробник подій `@bot.callback_query_handler`. Він відповідає за роботу із вбудованими (inline) кнопками. Після того, як користувач натисне на одну з них, робота з ботом продовжиться. За допомогою декоратора помічник запропонує обрати школу для перегляду та вибору. Адреси шкіл виглядають як звичайні кнопки під рядком для введення повідомлення (рис.3.13).

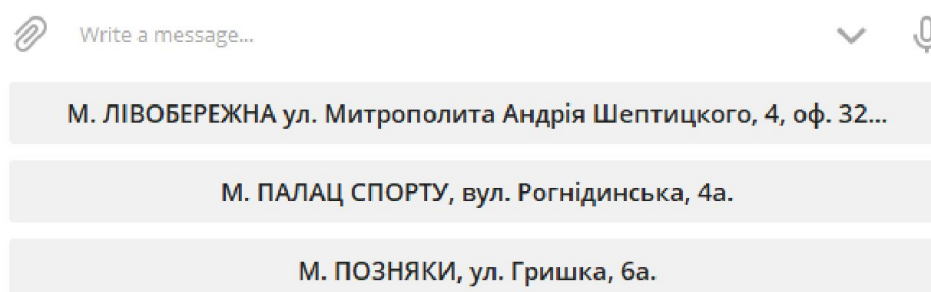


Рис.3.13. Вибір школи для запису

Як і у випадку з вибором курсів, є можливість перегляду всіх можливих шкіл з їх детальною інформацією, а саме: графіком роботи, розташуванням на карті та телефонами для зв'язку. Фрагмент коду, який описує створення та логіку роботи вбудованих кнопок з детальною інформацією зображено на

рисунку 3.14.

```
@bot.message_handler(func=lambda mess: mess.text=='М. ЛІВОБЕРЕЖНА ул. Митрополита Андрія Шептицького,
def choose_adress_school2(message):
    school_adress = message.text
    if school_adress == adress_Standart_1:
        bot.send_message(message.chat.id, adress_Standart_1)
        keyboard = types.InlineKeyboardMarkup()
        button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time1_1_1')
        button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Standart_url_1)
        button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone1_1_1')
        button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_16')
        keyboard.add(button_1)
        keyboard.add(button_2)
        keyboard.add(button_3)
        keyboard.add(button_4)
        bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
```

Рис.3.14. Створення кнопок детальної інформації

Параметр *callback\_data* визначає дані, які відправляються в *callback\_query*, що в свою чергу є об'єктом вхідного запиту зворотного зв'язку від вбудованої кнопки. Параметр *url* вказує на те, що кнопка з посиланням (в даному випадку на карти з місцем розміщення школи). Після натиску на таку кнопку, Телеграм запитує, чи варто відкривати це посилання. Якщо користувач згоден, то розміщення школи на карті відкривається у Google Maps (рис.3.15).

Графік роботи школи та телефони для зв'язку відображено як повідомлення від бота, яке він відсилає користувачеві після натискання на відповідні кнопки (рис.3.16).

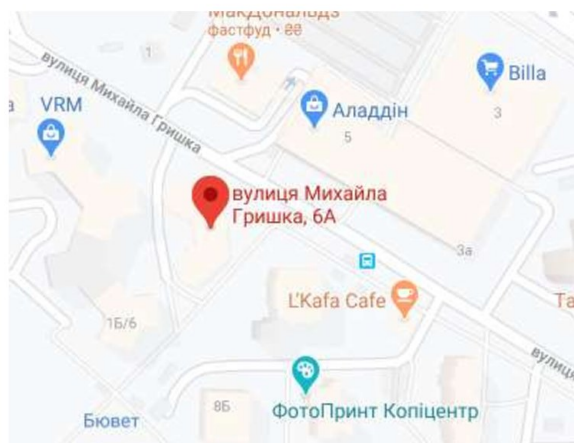


Рис.3.15. Розміщення школи на карті

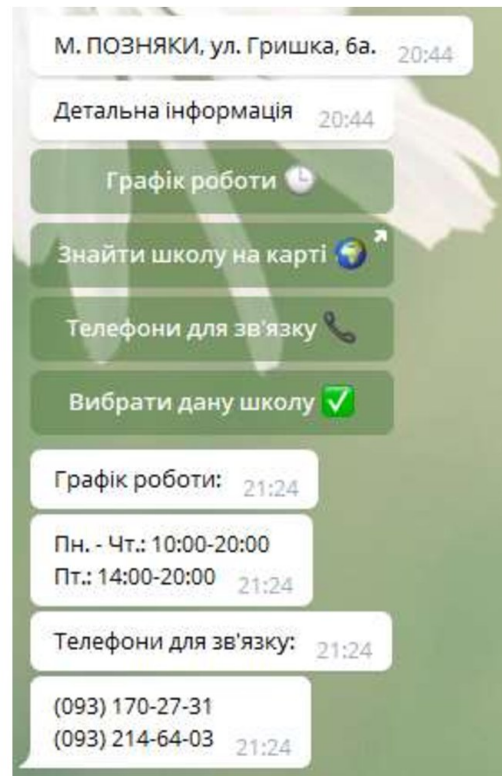


Рис.3.16. Реалізація та робота кнопок для детальної інформації у Telegram

Після вибору школи чат-бот запитує, який час є найбільш зручним для потенційного студента. Кнопки створюються як звичайна клавіатура аналогічно описаним вище прикладам (рис.3.17).

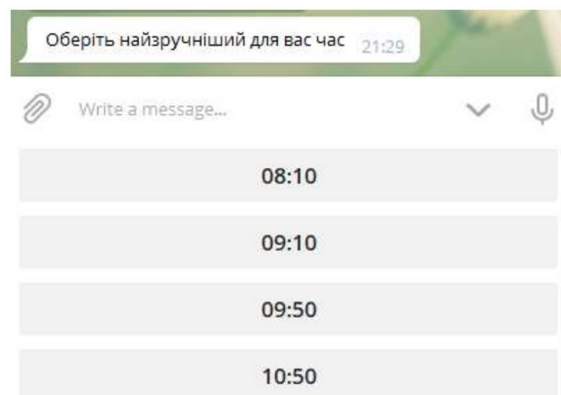


Рис.3.17. Вибір часу

Наступним кроком є введення даних користувачем, таких як: ПІБ, номер телефону, рівень англійської. Для останнього потрібно пройти тест, якщо користувач не впевнений у своїх знаннях. Це значення вибирається з

клавіатури, після чого чат-бот повідомляє про успішний запис на курс повідомленням (рис.3.18). Нижче знаходяться вбудовані кнопки з розміщенням школи на карті та умовами оплати курсу. Всі дані записуються до БД.

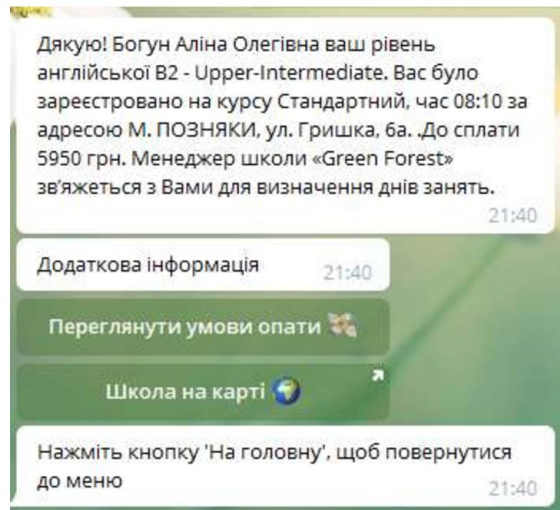


Рис.3.18. Результат після запису на курс

Запис на пробне заняття відбувається аналогічно. Єдиною відмінністю є те, що час вибирати не потрібно, оскільки для цього студенту зателефонує менеджер школи.

### *Розділ «Розклад розмовних клубів»*

Він складається зі списку розмовних клубів в обраному місті, у якому зазначено вид клубу, час та місце його проведення. Має вигляд повідомлення від бота у чаті. Бонусом є посилання на загальний Телеграм-канал для всіх клубів визначеного міста (рис.3.19). Дані беруться із файлу inform.py.

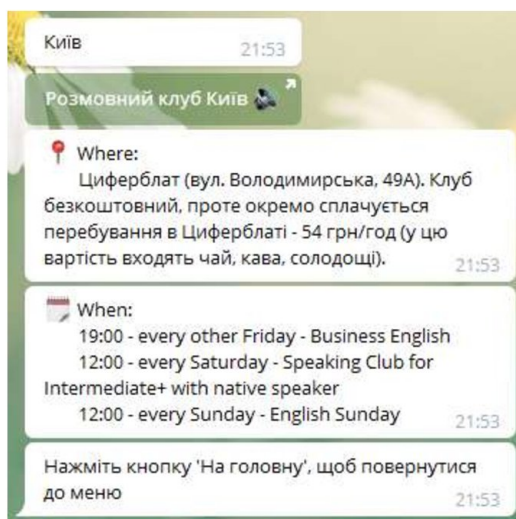


Рис.3.19. Розділ «Розклад розмовних клубів»

### *Розділ «Контакти»*

У цьому розділі знаходяться всі контакти школи «Green Forest»: офіційний сайт, Facebook, Instagram, YouTube, Telegram, Pinterest та CALL CENTER. Всі вони є inline-кнопками, більшість – з url посиланням (рис.3.20).



Рис.3.20. Розділ «Контакти»

### *Розділ «Найпоширеніші запитання»*

Цей розділ уміщує в себе відповіді на запитання, які найчастіше ставляться менеджерам школи. Ботом у чат після натискання на кнопку розділу відправляється посилання на Telegraph.

Telegraph – це веб-сервіс для створення та редагування статей від

команди Telegram. Редактор має наступні можливості:

- розширене форматування;
- можливість поширення статті за межами Telegram;
- вставка відео, гіф-зображень та звичайних зображень в текст;
- редагування тексту (шрифт, гіперпосилання, 2 види заголовків, цитування).

Для створення статті потрібно перейти на сайт telegraph.ph та слідувати інструкціям. Є можливість управління статтями через бот Telegraph у Telegram. Після створення публікації посилання на неї копіюється та вставляється у даному дипломному проекті в код чат-бота.

Відкриваються статті Telegraph за допомогою функції Телеграму Instant View. Вона дозволяє відкривати посилання на статті та веб-сайти не переходячи у браузер, а безпосередньо у Телеграм (для користувачів смартфонів). На рисунку 3.21 зображено використання Telegraph у чат-боті.

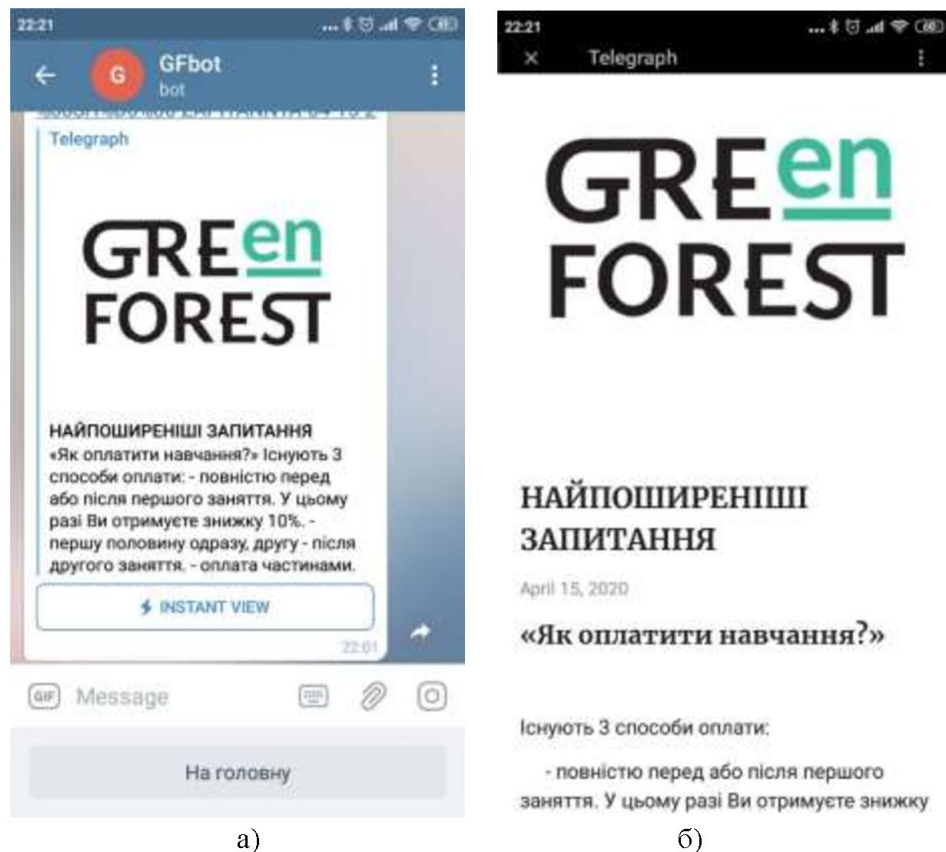


Рис.3.21. Розділ «Найпоширеніші запитання»: а) вигляд статті Telegraph у чаті; б) вигляд

статті після її відкриття за допомогою Instant View

### Розділ «Замовити дзвінок»

Даний розділ потрібен для того, щоб менеджери школи «Green Forest» зв'язалися з користувачем (рис.3.22). Все, що потрібно зробити – ввести ПІБ, номер телефону та коментар за необхідності (вказати час, коли зручно зателефонувати, або тему звернення). Дані аналогічно до функції *Записатися на курс/пробне заняття* записуються у БД.

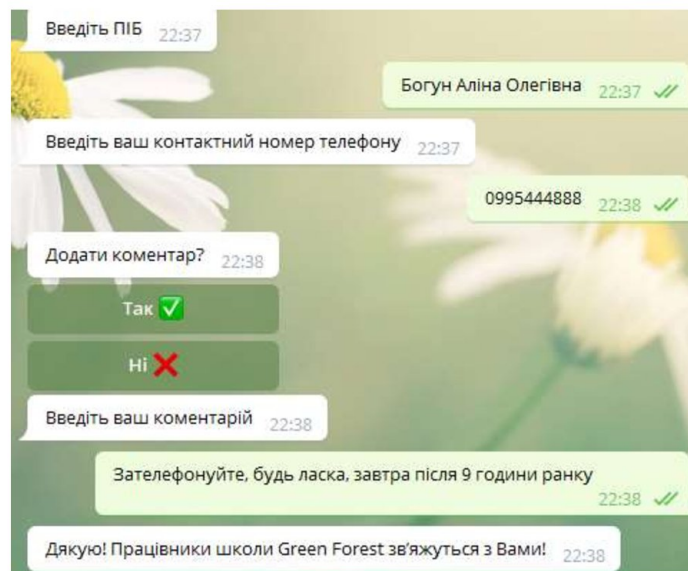


Рис.3.22. Розділ «Замовити дзвінок»

### 3.2.3 Підключення бази даних MySQL

Для збереження даних користувачів та подальшої роботи з ними, підключено MySQL. База даних потрібна для запису на курс/пробне заняття та замовлення дзвінка. Перш за все, встановлено *mysql.connector*, що дозволяє підключитися до створеного сервера MySQL.

У файлі *Створення бд.py* прописані всі необхідні команди для створення:

- з'єднання з БД за допомогою метода *connect()*;
- бази даних, використовуючи метод *execute()* з параметром *'CREATE DATABASE GF\_English\_School'*;

- курсора за допомогою метода *cursor()* (курсор – це спеціальний об’єкт, який виконує запити та отримує їх результати, тобто за його допомогою створюється порожня БД, записуються та редагуються дані);
- таблиць *enroll\_in\_a\_course*, *trial\_lesson* та *request\_a\_call* за допомогою метода *execute()*.

Таблиця *enroll\_in\_a\_course* має наступні поля: *id*, *City*, *Adress\_school*, *Name\_Surname*, *Phone\_number*, *Course*, *Time\_lesson*, *English\_skill*, *Price*, *Date*, *Time*.

Таблиця *trial\_lesson* має наступні поля: *id*, *City*, *Adress\_school*, *Name\_Surname*, *Phone\_number*, *Course*, *English\_skill*, *Date*, *Time*.

Таблиця *request\_a\_call* має наступні поля: *id*, *City*, *Name\_Surname*, *Phone\_number*, *Comment*, *Date*, *Time*.

Для того, щоб чат-бот не просто отримував дані від користувачів, а ще й записував їх у створену БД, у головному файлі програми відбувається підключення чат-бота до сервера та занесення даних за допомогою команди “*INSERT INTO*” та методу *execute()* (рис.3.23 та 3.24).

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="myfirstsqlserver228AB",
    auth_plugin='mysql_native_password',
    database="GF_English_School")
my_cursor = mydb.cursor()
```

Рис.3.23. Підключення до сервера з існуючою БД

```
registration_course = "INSERT INTO enroll_in_a_course (City, " \
    "Adress_school, Name_Surname,Phone_number, " \
    "Course, Time_lesson, English_skill, Price, Date, " \
    "Time) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
record1 = (city, adress, information, phone_number_registration, course,
    time_lesson, english_skill, price, date_now, time_now)
my_cursor.execute(registration_course, record1)
mydb.commit()
mydb.close()
```

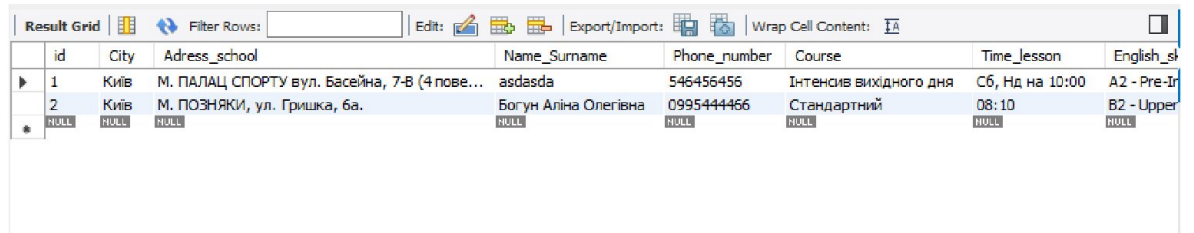
3.24. Запис даних у таблицю *enroll\_in\_course*

Після запису всіх даних до таблиці, зберігаємо зміни та закриваємо



з'єднання за допомогою методів *commit()* та *close()* відповідно.

Результат запису на курс стандартний зображено на рисунку 3.25.



The screenshot shows a 'Result Grid' window with a toolbar at the top. The toolbar includes a 'Filter Rows' field, an 'Edit' button, an 'Export/Import' button, and a 'Wrap Cell Content' checkbox. The grid contains two data rows and a summary row. The first row has id=1, City=Київ, Address\_school=М. ПАЛАЦ СПОРТУ вул. Басейна, 7-В (4 пове..., Name\_Surname=asdasda, Phone\_number=546456456, Course=Інтенсив вихідного дня, Time\_lesson=Сб, Нд на 10:00, and English\_sl=A2 - Pre-І. The second row has id=2, City=Київ, Address\_school=М. ПОЗНЯКИ, ул. Гришка, 6а., Name\_Surname=Богун Аліна Олегівна, Phone\_number=0995444466, Course=Стандартний, Time\_lesson=08:10, and English\_sl=B2 - Upper. The summary row shows a star icon and NULL values for all columns.

	id	City	Adress_school	Name_Surname	Phone_number	Course	Time_lesson	English_sl
▶	1	Київ	М. ПАЛАЦ СПОРТУ вул. Басейна, 7-В (4 пове...	asdasda	546456456	Інтенсив вихідного дня	Сб, Нд на 10:00	A2 - Pre-І
	2	Київ	М. ПОЗНЯКИ, ул. Гришка, 6а.	Богун Аліна Олегівна	0995444466	Стандартний	08:10	B2 - Upper
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.25. Дані у БД після запису на курс

## ВИСНОВКИ

Результатом виконання дипломного проекту є чат-бот мережі шкіл англійської мови «Green Forest» для Telegram.

В першому розділі досліджено предметну область, а саме платформу Telegram, чат-боти та загальні принципи їх роботи. Також розглянуто існуючі чат-боти для визначення теми та функціональності власного Telegram-помічника.

У другому розділі описано призначення чат-бота, що розроблявся, вимоги до нього та функції, які він повинен виконувати. Важливою частиною є також вибір мови програмування та середовища для розробки, бази даних та огляд інтерфейсу для роботи з ботами TelegramBotAPI. Правильний вибір всіх перерахованих складових безсумнівно впливає на роботу проекту в цілому.

У третьому розділі відображені етапи розробки чат-бота, починаючи від його створення в Telegram та закінчуючи оглядом його роботи з базою даних MySQL.

Завдяки правильному вибору всіх аспектів, необхідних для отримання якісного та актуального програмного продукту, розроблено чат-бот, що повністю відповідає описаним вимогам та функціональним характеристикам. Бот працює безперебійно з можливістю приймати запити від багатьох користувачів одночасно.

Чат-бот є унікальним помічником, який без сумніву оптимізуватиме роботу менеджерів школи, пришвидшить її без будь-яких небажаних наслідків для компанії в цілому.

Проект має перспективу розширювати свої можливості та розповсюджуватися на ринку IT-технологій. Найважливішим кроком, що очікується у майбутньому, є завантаження чат-бота на сервер для цілодобової роботи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чат-бот - что это такое, как работает, виды, преимущества и недостатки, как создать [Электронный ресурс] – Режим доступа: <http://kak-bog.ru/chat-bot-cto-eto-takoe-kak-rabotaet-vidy-preimushchestva>
2. Чат-боты: обзор и состояние технологий в отрасли [Электронный ресурс] – Режим доступа: <http://nlpx.net/archives/425>
3. 50 лучших чат-ботов для Telegram [Электронный ресурс] – Режим доступа: <https://uip.me/2016/04/50-popular-telegram-bots/>
4. 2018 State of Chatbots Report [Электронный ресурс] – Режим доступа до ресурсу: <https://www.drift.com/>
5. Telegram Bot API [Электронный ресурс] – Режим доступа до ресурсу: <https://core.telegram.org/bots/api>.
6. Самоучитель Python [Электронный ресурс] – Режим доступа до ресурсу: <https://pythonworld.ru/samouchitel-python>.
7. Ларман К. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание / Крэг Ларман; ООО “И.Д. Вильямс”, 2013. – 736 с.
8. Telegram FAQ [Электронный ресурс] – Режим доступа до ресурсу: <https://telegram.org/>.
9. pyTelegramBotAPI [Электронный ресурс]-Режим доступа: <https://github.com/eternnoir/pyTelegramBotAPI>
10. Как создать Telegram бота с помощью Python [Электронный ресурс] – Режим доступа: <https://medium.com/nuances-of-programming/>
11. Офіційний сайт MySQL [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mysql.com/>.
12. Офіційний сайт "Green Forest" [Электронный ресурс] – Режим доступа до ресурсу: <https://greenforest.com.ua/>.

## ДОДАТОК А. ТЕКСТ ПРОГРАМИ

### Файл Телеграм бот.ру

```

#t.me/GF100_bot
import mysql.connector
import telebot
from telebot import types
import datetime
import const
from inform import token, hello, save_answer, choose_city, city_1, city_2, city_3, city_4, city_5, Choose_course,
Talk_Club_Schedule, Frequently_Asked_Questions, get_call, contacts, back_city_selection, text_choose_course,
Intense_Weekend_text, Intense_text, Standard_text, on_main_text, main_page, back_choose_city, make_choice,
enroll_course, enroll_trial_lesson, Additional_Information, detailed_information, Work_Schedule,
Find_school_map, Communication_Phones, Select_school, choose_school_text, Work_Schedule_text,
Communication_Phones_text, Choose_time_text, Intense_Weekend, price_Intense_Weekend,
adress_Intense_Weekend_1, Intense_Weekend_url_1, adress_Intense_Weekend_2, Intense_Weekend_url_2,
adress_Intense_Weekend_3, Intense_Weekend_url_3, adress_Intense_Weekend_4, Intense_Weekend_url_4,
adress_Intense_Weekend_5, Intense_Weekend_url_5, adress_Intense_Weekend_6, Intense_Weekend_url_6,
Intense_Weekend_Work_Schedule_1, Intense_Weekend_Work_Schedule_2,
Intense_Weekend_Work_Schedule_3_0, Intense_Weekend_Work_Schedule_3_1,
Intense_Weekend_Work_Schedule_4_0, Intense_Weekend_Work_Schedule_4_1,
Intense_Weekend_Work_Schedule_5,
Intense_Weekend_Work_Schedule_6, Intense_Weekend_Communication_Phones_1_0, Intense_Weekend_Communi
cation_Phones_1_1, Intense_Weekend_Communication_Phones_1_2,
Intense_Weekend_Communication_Phones_2_0, Intense_Weekend_Communication_Phones_2_1,
Intense_Weekend_Communication_Phones_2_2, Intense_Weekend_Communication_Phones_3_0,
Intense_Weekend_Communication_Phones_3_1, Intense_Weekend_Communication_Phones_4_0,
Intense_Weekend_Communication_Phones_4_1, Intense_Weekend_Communication_Phones_5_0,
Intense_Weekend_Communication_Phones_5_1, Intense_Weekend_Communication_Phones_5_2,
Intense_Weekend_Communication_Phones_6_0, Intense_Weekend_Communication_Phones_6_1,
Intense_Weekend_Choose_time_1_0, Intense_Weekend_Choose_time_1_1, Intense_Weekend_Choose_time_2_0,
Intense_Weekend_Choose_time_2_1, Intense_Weekend_Choose_time_3_0, Intense_Weekend_Choose_time_4_0,
Intense_Weekend_Choose_time_5_0, Intense_Weekend_Choose_time_6_0, Intense_Weekend_Choose_time_6_1,
Intense, price_Intense, adress_Intense_1, adress_Intense_2, adress_Intense_3, Intense_url_1, Intense_url_2,
Intense_url_3, adress_Intense_4, Intense_url_4, adress_Intense_5, adress_Intense_6, adress_Intense_7, Intense_url_6,
adress_Intense_7, Intense_url_7, adress_Intense_8, adress_Intense_9, Intense_url_8, Intense_url_9,
Intense_Work_Schedule_1_0, Intense_Work_Schedule_1_1, Intense_Work_Schedule_1_2,
Intense_Work_Schedule_2_0, Intense_Work_Schedule_2_1, Intense_Work_Schedule_3_0,
Intense_Work_Schedule_3_1, Intense_Work_Schedule_4, Intense_Work_Schedule_5_0,
Intense_Work_Schedule_5_1, Intense_Work_Schedule_6_0, Intense_Work_Schedule_6_1,
Intense_Work_Schedule_7, Intense_Work_Schedule_8, Intense_Work_Schedule_9,
Intense_Communication_Phones_1_0, Intense_Communication_Phones_1_1,
Intense_Communication_Phones_1_2, Intense_Communication_Phones_2_0,
Intense_Communication_Phones_2_1, Intense_Communication_Phones_2_2, Intense_Communication_Phones_2_3,
Intense_Communication_Phones_3_0, Intense_Communication_Phones_3_1,
Intense_Communication_Phones_4_0, Intense_Communication_Phones_4_1,
Intense_Communication_Phones_4_2, Intense_Communication_Phones_5_0,
Intense_Communication_Phones_5_1, Intense_Communication_Phones_6_0,
Intense_Communication_Phones_6_1, Intense_Communication_Phones_7_0,
Intense_Communication_Phones_7_1, Intense_Communication_Phones_7_2,
Intense_Communication_Phones_8_0, Intense_Communication_Phones_8_1,
Intense_Communication_Phones_9_0, Intense_Communication_Phones_9_1,
Intense_Communication_Phones_9_2,
Intense_Choose_time_1_0, Intense_Choose_time_1_1, Intense_Choose_time_1_2, Intense_Choose_time_1_3, Intense
_Choose_time_1_4, Intense_Choose_time_1_5, Intense_Choose_time_2_0, Intense_Choose_time_2_1,
Intense_Choose_time_2_2, Intense_Choose_time_2_3, Intense_Choose_time_2_4, Intense_Choose_time_2_5,
Intense_Choose_time_3_0, Intense_Choose_time_3_1, Intense_Choose_time_3_2, Intense_Choose_time_3_3,
Intense_Choose_time_3_4, Intense_Choose_time_3_5, Intense_Choose_time_4_0, Intense_Choose_time_4_1,
Intense_Choose_time_4_2, Intense_Choose_time_4_3, Intense_Choose_time_4_4, Intense_Choose_time_5_0,
Intense_Choose_time_5_1, Intense_Choose_time_5_2, Intense_Choose_time_5_3, Intense_Choose_time_5_4,
Intense_Choose_time_6_0, Intense_Choose_time_6_1, Intense_Choose_time_6_2, Intense_Choose_time_6_3,
Intense_Choose_time_6_4, Intense_Choose_time_7_0, Intense_Choose_time_7_1, Intense_Choose_time_7_2,
Intense_Choose_time_7_3, Intense_Choose_time_7_4, Intense_Choose_time_8_0, Intense_Choose_time_8_1,
Intense_Choose_time_8_2, Intense_Choose_time_8_3, Intense_Choose_time_8_4, Intense_Choose_time_9_0,

```

Intense\_Choose\_time\_9\_1, Intense\_Choose\_time\_9\_2, Intense\_Choose\_time\_9\_3, Intense\_Choose\_time\_9\_4,  
 Standart, price\_Standart, adress\_Standart\_1, adress\_Standart\_2, adress\_Standart\_3, Standart\_url\_1, Standart\_url\_2,  
 Standart\_url\_3, adress\_Standart\_4, Standart\_url\_4, adress\_Standart\_5, Standart\_url\_5, adress\_Standart\_6,  
 Standart\_url\_6, adress\_Standart\_7, Standart\_url\_7, adress\_Standart\_8, adress\_Standart\_9, Standart\_url\_8,  
 Standart\_url\_9, Standart\_Work\_Schedule\_1\_0, Standart\_Work\_Schedule\_1\_1, Standart\_Work\_Schedule\_1\_2,  
 Standart\_Work\_Schedule\_2\_0, Standart\_Work\_Schedule\_3\_0, Standart\_Work\_Schedule\_3\_1,  
 Standart\_Work\_Schedule\_4, Standart\_Work\_Schedule\_5\_0, Standart\_Work\_Schedule\_5\_1,  
 Standart\_Work\_Schedule\_6\_0, Standart\_Work\_Schedule\_6\_1, Standart\_Work\_Schedule\_7,  
 Standart\_Work\_Schedule\_8, Standart\_Work\_Schedule\_9, Standart\_Communication\_Phones\_1\_0,  
 Standart\_Communication\_Phones\_1\_1, Standart\_Communication\_Phones\_1\_2,  
 Standart\_Communication\_Phones\_2\_0, Standart\_Communication\_Phones\_2\_1,  
 Standart\_Communication\_Phones\_3\_0, Standart\_Communication\_Phones\_3\_1,  
 Standart\_Communication\_Phones\_4\_0, Standart\_Communication\_Phones\_4\_1,  
 Standart\_Communication\_Phones\_4\_2, Standart\_Communication\_Phones\_5\_0,  
 Standart\_Communication\_Phones\_5\_1, Standart\_Communication\_Phones\_6\_0,  
 Standart\_Communication\_Phones\_6\_1, Standart\_Communication\_Phones\_7\_0,  
 Standart\_Communication\_Phones\_7\_1, Standart\_Communication\_Phones\_7\_2,  
 Standart\_Communication\_Phones\_8\_0, Standart\_Communication\_Phones\_8\_1,  
 Standart\_Communication\_Phones\_8\_2, Standart\_Communication\_Phones\_8\_3,  
 Standart\_Communication\_Phones\_9\_0, Standart\_Communication\_Phones\_9\_1,  
 Standart\_Communication\_Phones\_9\_2, Standart\_Choose\_time\_1\_0, Standart\_Choose\_time\_1\_1,  
 Standart\_Choose\_time\_1\_2, Standart\_Choose\_time\_1\_3, Standart\_Choose\_time\_1\_4, Standart\_Choose\_time\_1\_5,  
 Standart\_Choose\_time\_1\_6, Standart\_Choose\_time\_1\_7, Standart\_Choose\_time\_2\_0, Standart\_Choose\_time\_2\_1,  
 Standart\_Choose\_time\_2\_2, Standart\_Choose\_time\_2\_3, Standart\_Choose\_time\_2\_4, Standart\_Choose\_time\_2\_5,  
 Standart\_Choose\_time\_2\_6, Standart\_Choose\_time\_2\_7, Standart\_Choose\_time\_3\_0, Standart\_Choose\_time\_3\_1,  
 Standart\_Choose\_time\_3\_2, Standart\_Choose\_time\_3\_3, Standart\_Choose\_time\_3\_4, Standart\_Choose\_time\_3\_5,  
 Standart\_Choose\_time\_3\_6, Standart\_Choose\_time\_3\_7, Standart\_Choose\_time\_4\_0, Standart\_Choose\_time\_4\_1,  
 Standart\_Choose\_time\_4\_2, Standart\_Choose\_time\_4\_3, Standart\_Choose\_time\_4\_4, Standart\_Choose\_time\_4\_5,  
 Standart\_Choose\_time\_4\_6, Standart\_Choose\_time\_5\_0, Standart\_Choose\_time\_5\_1, Standart\_Choose\_time\_5\_2,  
 Standart\_Choose\_time\_5\_3, Standart\_Choose\_time\_5\_4, Standart\_Choose\_time\_5\_5, Standart\_Choose\_time\_6\_0,  
 Standart\_Choose\_time\_6\_1, Standart\_Choose\_time\_6\_2, Standart\_Choose\_time\_6\_3, Standart\_Choose\_time\_6\_4,  
 Standart\_Choose\_time\_6\_5, Standart\_Choose\_time\_7\_0, Standart\_Choose\_time\_7\_1, Standart\_Choose\_time\_7\_2,  
 Standart\_Choose\_time\_7\_3, Standart\_Choose\_time\_7\_4, Standart\_Choose\_time\_8\_0, Standart\_Choose\_time\_8\_1,  
 Standart\_Choose\_time\_8\_2, Standart\_Choose\_time\_8\_3, Standart\_Choose\_time\_8\_4, Standart\_Choose\_time\_8\_5,  
 Standart\_Choose\_time\_9\_0, Standart\_Choose\_time\_9\_1, Standart\_Choose\_time\_9\_2, Standart\_Choose\_time\_9\_3,  
 Standart\_Choose\_time\_9\_4, Standart\_Choose\_time\_9\_5, Full\_Name\_Surname, phone\_number\_text,  
 english\_test\_text, test\_text, url\_test, english\_level\_text, registratio\_course\_1, pay, url\_pay, map\_school,  
 english\_level, A1, A2, B1, B2, C1, registratio\_course\_2, call\_text, comment\_yes, comment\_no, add\_comment\_text,  
 write\_comment, end\_call\_a, choose\_school\_info, Standart\_text\_1, payment, where\_Kiev, when\_Kiev,  
 where\_Dnipro, when\_Dnipro, where\_Xarkiv, when\_Xarkiv, where\_Odesa, when\_Odesa, when\_Lviv, when\_Lviv,  
 club\_text\_Kiev, club\_text\_Dnipro, club\_text\_Xarkiv, club\_text\_Odesa, club\_text\_Lviv, url\_Kiev, url\_Dnipro,  
 url\_Xarkiv, url\_Odesa, url\_Lviv, url\_questions, web\_cite, Facebook, Instagram, YouTube, Telegram, Pinterest,  
 CALL\_CENTER, url\_web\_cite, url\_Facebook, url\_Instagram, url\_YouTube, url\_Telegram, url\_Pinterest,  
 CALL\_CENTER\_1, CALL\_CENTER\_2, CALL\_CENTER\_3, CALL\_CENTER\_4, questions\_text

bot = telebot.TeleBot(token)

*# Початок роботи бота /start or /help. Вибір міста*

@bot.message\_handler(commands=['help', 'start'])

def send\_welcome(message):

markup = types.ReplyKeyboardMarkup(resize\_keyboard=True, row\_width=2)

itembtn1 = types.KeyboardButton(city\_1)

itembtn2 = types.KeyboardButton(city\_2)

itembtn3 = types.KeyboardButton(city\_3)

itembtn4 = types.KeyboardButton(city\_4)

itembtn5 = types.KeyboardButton(city\_5)

markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5)

bot.send\_message(message.chat.id, hello, reply\_markup=markup)

*# Після вибору міста, створення наступних кнопок. Обрати курс, Розклад розмовних клубів...*

@bot.message\_handler(func=lambda mess: mess.text=='Київ' or mess.text=='Дніпро' or mess.text=='Харків' or  
 mess.text=='Одеса' or mess.text=='Львів' and mess.content\_type=='text')

def first\_menu(message):

global city

```

city = str(message.text)
markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
itembtn1 = types.KeyboardButton(Choose_course)
itembtn2 = types.KeyboardButton(Talk_Club_Schedule)
itembtn3 = types.KeyboardButton(Frequently_Asked_Questions)
itembtn4 = types.KeyboardButton(get_call)
itembtn5 = types.KeyboardButton(contacts)
itembtn6 = types.KeyboardButton(back_city_selection)
markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5, itembtn6)
bot.send_message(message.chat.id, save_answer, reply_markup=markup)

# Кнопка обрати курс. Створення наступних кнопок: Інтенсив вихідного дня, Інтенсивний курс...
@bot.message_handler(func=lambda mess: mess.text=='Обрати курс' and mess.content_type=='text')
def choose_course(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton(Intense_Weekend_text)
    itembtn2 = types.KeyboardButton(Intense_text)
    itembtn3 = types.KeyboardButton(Standard_text)
    itembtn4 = types.KeyboardButton(main_page)
    markup.add(itembtn1, itembtn2, itembtn3, itembtn4)
    bot.send_message(message.chat.id, text_choose_course, reply_markup=markup)

# Курс Інтенсив вихідного дня. Створення кнопок: Записатися на курс, Записатися на пробне заняття
@bot.message_handler(func=lambda mess: mess.text=='Інтенсив вихідного дня' and mess.content_type=='text')
def Intense_Weekend_course(message):
    bot.send_message(message.chat.id, Intense_Weekend)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=enroll_course, callback_data='button1_1')
    button_2 = types.InlineKeyboardButton(text=enroll_trial_lesson, callback_data='button2_1')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(message.chat.id, make_choice, reply_markup=keyboard)

# Курс Інтенсивний. Створення кнопок: Записатися на курс, Записатися на пробне заняття
@bot.message_handler(func=lambda mess: mess.text=='Інтенсивний курс' and mess.content_type=='text')
def Intense_course(message):
    bot.send_message(message.chat.id, Intense)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=enroll_course, callback_data='button3_1')
    button_2 = types.InlineKeyboardButton(text=enroll_trial_lesson, callback_data='button4_1')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(message.chat.id, make_choice, reply_markup=keyboard)

# Курс Стандартний. Створення кнопок: Записатися на курс, Записатися на пробне заняття
@bot.message_handler(func=lambda mess: mess.text=='Стандартний курс' and mess.content_type=='text')
def Standard_course(message):
    bot.send_message(message.chat.id, Standard)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=enroll_course, callback_data='button5_1')
    button_2 = types.InlineKeyboardButton(text=enroll_trial_lesson, callback_data='button6_1')
    keyboard.add(button_1)
    keyboard.add(button_2)
    bot.send_message(message.chat.id, make_choice, reply_markup=keyboard)

#ІНТЕНСИВ ВИХІДНОГО ДНЯ. ЗАПИСАТИСЯ НА КУРС, ЗАПИСАТИСЯ НА ПРОБНЕ ЗАНЯТТЯ
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('button1_1') or
c.data.startswith('button2_1'))
def button1_1(c):
    global course, button_check
    course = Intense_Weekend_text
    button_check = str(c.data)
    if city == city_1:
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        itembtn1 = types.KeyboardButton(adress_Intense_Weekend_1)
        markup.add(itembtn1)

```

```

bot.send_message(c.message.chat.id, choose_school_info, reply_markup=markup)
elif city == city_2:
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton(adress_Intense_Weekend_2)
    markup.add(itembtn1)
    bot.send_message(c.message.chat.id, choose_school_info, reply_markup=markup)
elif city == city_3:
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton(adress_Intense_Weekend_3)
    itembtn2 = types.KeyboardButton(adress_Intense_Weekend_4)
    markup.add(itembtn1)
    markup.add(itembtn2)
    bot.send_message(c.message.chat.id, choose_school_info, reply_markup=markup)
elif city == city_4:
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton(adress_Intense_Weekend_5)
    markup.add(itembtn1)
    bot.send_message(c.message.chat.id, choose_school_info, reply_markup=markup)
elif city == city_5:
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton(adress_Intense_Weekend_6)
    markup.add(itembtn1)
    bot.send_message(c.message.chat.id, choose_school_info, reply_markup=markup)

```

*# Для адресів школи створення кнопок: Графік роботи, Знайти школу на карті...*

```

@bot.message_handler(func=lambda mess: mess.text=='М. ПАЛАЦ СПОРТУ вул. Басейна, 7-В (4
поверх).' or mess.text=='вул. Європейська (колишня вул. Миронова), 5 (2 поверх).' or mess.text=='М.
АРХІТЕКТОРА БЕКЕТОВА вул. Дарвіна, 4, поверх 1 (вхід з двору).' or mess.text=='М. ІСТОРИЧНИЙ
МУЗЕЙ, вул. Римарська, 32.' or mess.text=='вул. Тираспольська, 4, 3 поверх.' or mess.text=='вул.
Данилишина, 6 (офісний центр "The Office", 3 поверх).' and mess.content_type=='text')

```

```

def choose_address_school(message):
    school_address = message.text
    if school_address == adress_Intense_Weekend_1:
        bot.send_message(message.chat.id, adress_Intense_Weekend_1)
        keyboard = types.InlineKeyboardMarkup()
        button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_1')
        button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_1)
        button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_1')
        button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_1')
        keyboard.add(button_1)
        keyboard.add(button_2)
        keyboard.add(button_3)
        keyboard.add(button_4)
        bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
    elif school_address == adress_Intense_Weekend_2:
        bot.send_message(message.chat.id, adress_Intense_Weekend_2)
        keyboard = types.InlineKeyboardMarkup()
        button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_2')
        button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_2)
        button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_2')
        button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_2')
        keyboard.add(button_1)
        keyboard.add(button_2)
        keyboard.add(button_3)
        keyboard.add(button_4)
        bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
    elif school_address == adress_Intense_Weekend_3:
        bot.send_message(message.chat.id, adress_Intense_Weekend_3)
        keyboard = types.InlineKeyboardMarkup()
        button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_3')
        button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_3)
        button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_3')
        button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_3')

```

```

keyboard.add(button_1)
keyboard.add(button_2)
keyboard.add(button_3)
keyboard.add(button_4)
bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
elif school_address == adress_Intense_Weekend_4:
    bot.send_message(message.chat.id, adress_Intense_Weekend_4)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_4')
    button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_4)
    button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_4')
    button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_4')
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    keyboard.add(button_4)
    bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
elif school_address == adress_Intense_Weekend_5:
    bot.send_message(message.chat.id, adress_Intense_Weekend_5)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_5')
    button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_5)
    button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_5')
    button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_5')
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    keyboard.add(button_4)
    bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)
elif school_address == adress_Intense_Weekend_6:
    bot.send_message(message.chat.id, adress_Intense_Weekend_6)
    keyboard = types.InlineKeyboardMarkup()
    button_1 = types.InlineKeyboardButton(text=Work_Schedule, callback_data='time_6')
    button_2 = types.InlineKeyboardButton(text=Find_school_map, url=Intense_Weekend_url_6)
    button_3 = types.InlineKeyboardButton(text=Communication_Phones, callback_data='phone_6')
    button_4 = types.InlineKeyboardButton(text=Select_school, callback_data='choose_school_6')
    keyboard.add(button_1)
    keyboard.add(button_2)
    keyboard.add(button_3)
    keyboard.add(button_4)
    bot.send_message(message.chat.id, detailed_information, reply_markup=keyboard)

# Графік робіт різних шкіл Курсу Інтенсив вихідного дня
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_1'))
def time_1(c):
    bot.send_message(c.message.chat.id, Work_Schedule_text)
    bot.send_message(c.message.chat.id, Intense_Weekend_Work_Schedule_1)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_2'))
def time_2(c):
    bot.send_message(c.message.chat.id, Work_Schedule_text)
    bot.send_message(c.message.chat.id, Intense_Weekend_Work_Schedule_2)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_3'))
def time_3(c):
    bot.send_message(c.message.chat.id, Work_Schedule_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Work_Schedule_3_0+'\n'+Intense_Weekend_Work_Schedule_3_1)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_4'))
def time_4(c):
    bot.send_message(c.message.chat.id, Work_Schedule_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Work_Schedule_4_0+'\n'+Intense_Weekend_Work_Schedule_4_1)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_5'))
def time_5(c):

```



```

bot.send_message(c.message.chat.id, Work_Schedule_text)
bot.send_message(c.message.chat.id, Intense_Weekend_Work_Schedule_5)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('time_6'))
def time_6(c):
    bot.send_message(c.message.chat.id, Work_Schedule_text)
    bot.send_message(c.message.chat.id, Intense_Weekend_Work_Schedule_6)

# Телефони для зв'язку різних шкіл Курсу Інтенсив вихідного дня
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_1'))
def phone_1(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Communication_Phones_1_0+'\n'+Intense_Weekend_Communication_Phones_1_1+'\n'+Intense_Weekend_Communication_Phones_1_2)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_2'))
def phone_2(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Communication_Phones_2_0+'\n'+Intense_Weekend_Communication_Phones_2_1+'\n'+Intense_Weekend_Communication_Phones_2_2)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_3'))
def phone_3(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Communication_Phones_3_0+'\n'+Intense_Weekend_Communication_Phones_3_1)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_4'))
def phone_4(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id,
Intense_Weekend_Communication_Phones_4_0+'\n'+Intense_Weekend_Communication_Phones_4_1)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_5'))
def phone_5(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id, Intense_Weekend_Communication_Phones_5_0+'\n'+
Intense_Weekend_Communication_Phones_5_1+'\n'+Intense_Weekend_Communication_Phones_5_2)
@bot.callback_query_handler(func=lambda c: c.data and c.data.startswith('phone_6'))
def phone_6(c):
    bot.send_message(c.message.chat.id, Communication_Phones_text)
    bot.send_message(c.message.chat.id, Intense_Weekend_Communication_Phones_6_0
+'\n'+Intense_Weekend_Communication_Phones_6_1)

if __name__ == '__main__':
    bot.polling(none_stop=True)

```

### Файл Створення бд.ру

```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="myfirstsqlserver228AB",
    auth_plugin='mysql_native_password')

my_cursor = mydb.cursor()

my_cursor.execute('CREATE DATABASE GF_English_School')

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="myfirstsqlserver228AB",
    auth_plugin='mysql_native_password',
    database = "GF_English_School")

my_cursor = mydb.cursor()

my_cursor.execute("id INTEGER AUTO_INCREMENT PRIMARY KEY, City VARCHAR(255), Adress_school
VARCHAR(255), Name_Surname VARCHAR(255) CREATE TABLE enroll_in_a_course (id INTEGER
AUTO_INCREMENT PRIMARY KEY, City VARCHAR(255), Adress_school VARCHAR(255), Name_Surname
VARCHAR(255), Phone_number VARCHAR(255), Course VARCHAR(255), Time_lesson VARCHAR(255),
English_skill VARCHAR(255), Price VARCHAR(255), Date VARCHAR(255), Time VARCHAR(255))")

my_cursor.execute("CREATE TABLE trial_lesson (id INTEGER AUTO_INCREMENT PRIMARY KEY, City
VARCHAR(255), Adress_school VARCHAR(255), Name_Surname VARCHAR(255), Phone_number
VARCHAR(255), Course VARCHAR(255), Time_lesson VARCHAR(255), English_skill VARCHAR(255), Date
VARCHAR(255), Time VARCHAR(255))")

my_cursor.execute("CREATE TABLE request_a_call (id INTEGER AUTO_INCREMENT PRIMARY KEY, City
VARCHAR(255), Name_Surname VARCHAR(255), Phone_number VARCHAR(255), Comment VARCHAR(255),
Date VARCHAR(255), Time VARCHAR(255))")

mydb.close()
```

### Файл const.py

```
# Реєстрація на курс, пробне заняття
city = ""
english_skill = ""
time_lesson = ""
course = ""
adress = ""
information = ""
phone_number_registration = ""
url_map = ""
button_check = ""
price = ""
time_now = ""
date_now = ""
# Замовити дзвінок
pip_call = ""
phonenummer_call = ""
comment_call = ""
comment_add = ""
```

## Файл inform.py

```

from const import information, english_skill, course, time_lesson, adress, price
# Початок
token = '1282379875:AAGRROa79q8qWETCTgeqH0VE7M6w3ORURSo'
hello = "Привіт! Це школа англійської мови «Green Forest». Яке місто тебе цікавить?"
save_answer = "Дякую, я запам'ятав! Обери, яке питання тебе цікавить нижче."
choose_city = "Яке місто тебе цікавить?"

# Міста
city_1 = 'Київ'
city_2 = 'Дніпро'
city_3 = 'Харків'
city_4 = 'Одеса'
city_5 = 'Львів'

# Меню
Choose_course = 'Обрати курс'
Talk_Club_Schedule = 'Розклад розмовних клубів'
Frequently_Asked_Questions = 'Найпоширеніші запитання'
get_call = 'Замовити дзвінок'
contacts = 'Контакти'
back_city_selection = 'Повернутися до вибору міста'

# Обрати курс
text_choose_course = "Оберіть курс"
Intense_Weekend_text = 'Інтенсив вихідного дня'
Intense_text = 'Інтенсивний курс'
Standard_text = 'Стандартний курс'
Standard_text_1 = 'Стандартний'
# На головну, повернутися до вибору міста
on_main_text = "Нажміть кнопку 'На головну', щоб повернутися до меню"
main_page = 'На головну'
back_choose_city = 'Повернутися до вибору міста'

# Записатися на курс, записатися на пробне заняття
make_choice = "Зробіть свій вибір"
enroll_course = "Записатися на курс 
```

enroll\_trial\_lesson = "Записатися на пробне заняття

```

# Детальна інформація про школи англійської мови
Additional_Information = "Додаткова інформація"
detailed_information = "Детальна інформація"
Work_Schedule = "Графік роботи ☺"
Find_school_map = "Знайти школу на карті 📍"
Communication_Phones = "Телефони для зв'язку 📞"
Select_school = "Вибрати дану школу ✔"

choose_school_text = "Оберіть, яка саме школа вас цікавить"
Work_Schedule_text = "Графік роботи: "
Communication_Phones_text = "Телефони для зв'язку: "
Choose_time_text = "Оберіть найзручніший для вас час"
questions_text = "Обери, яке питання тебе цікавить нижче"
# Курс ІНТЕНСИВ ВИХІДНОГО ДНЯ
Intense_Weekend = ""
☛ІНТЕНСИВ ВИХІДНОГО ДНЯ
9 ТИЖНІВ
СТАРТ: 4.07

♥Ціна 4950 грн/семестр

■2 зан. / нед. * 3 г. 40 хв.

```

📅Графік: Сб. і Нд. на 10:00 / 14:10

1. ШВИДКО: цілий рівень за 9 тижнів
2. ТІЛЬКИ ПО ВИХІДНИХ: заняття в суботу та неділю по 4 години
3. ІННОВАЦІЙНО: навчання на базі унікальної гібридної платформи Notes by Green Forest
4. ЗРУЧНО: контроль за навчанням через додаток My Green Forest: домашні завдання, зміна графіка, успішність, новини, запис на додаткові заняття
5. ПРОФЕСІЙНО: всі викладачі Green Forest проходять підготовку в Green Forest Teaching Centre by Nadia Diachuk
6. БЕЗКОШТОВНІ БОНУСИ: онлайн версія підручника, розмовні клуби з носіями мови, спецкурси на різні граматичні та лексичні теми

price\_Intense\_Weekend = "4950 грн."

# Київ

adress\_Intense\_Weekend\_1 = 'М. ПАЛАЦ СПОРТУ вул. Басейна, 7-В (4 поверх).'

Intense\_Weekend\_url\_1 = "https://cutt.ly/mtIeM1S"

# Дніпро

adress\_Intense\_Weekend\_2 = 'вул. Європейська (колишня вул. Миронова), 5 (2 поверх).'

Intense\_Weekend\_url\_2 = "https://cutt.ly/FtOiSVC"

# Графік роботи

# Київ

Intense\_Weekend\_Work\_Schedule\_1 = "Пн. - Пт.: 14:00-20:00"

# Телефони для зв'язку

# Київ

Intense\_Weekend\_Communication\_Phones\_1\_0 = "(093) 170-27-36"

Intense\_Weekend\_Communication\_Phones\_1\_1 = "(093) 030-43-53"

Intense\_Weekend\_Communication\_Phones\_1\_2 = "(044) 383-95-05"

# Час заняття

# Київ

Intense\_Weekend\_Choose\_time\_1\_0 = 'Сб, Нд на 10:00'

Intense\_Weekend\_Choose\_time\_1\_1 = 'Сб, Нд на 14:10'

# Курс ІНТЕНСИВНИЙ

Intense = ""

🎯КУРС ІНТЕНСИВНИЙ

6 ТИЖНІВ

СТАРТ: 15.06 | 27.07

💎Ціна 4950 грн/семестр

📅4 зан. / Тиждень. \* 1 ч. 50 хв.

📅Графік: Пн.- Чт. на 8:10 / 10:10 / 12:00 / 16:10 / 18:10 / 20:10

1. ШВИДКО: цілий рівень за 9 тижнів
2. НАСИЧЕНО: 4 заняття в тиждень по 2 години
3. ІННОВАЦІЙНО: навчання на базі унікальної гібридної платформи Notes by Green Forest
4. ЗРУЧНО: контроль за навчанням через додаток My Green Forest: домашні завдання, зміна графіка, успішність, новини, запис на додаткові заняття
5. ПРОФЕСІЙНО: всі викладачі Green Forest проходять підготовку в Green Forest Teaching Centre by Nadia Diachuk
6. БЕЗКОШТОВНІ БОНУСИ: онлайн версія підручника, розмовні клуби з носіями мови, спецкурси на різні граматичні та лексичні теми

price\_Intense = "4950 грн."

# Київ

adress\_Intense\_1 = 'М. ЛІВОБЕРЕЖНА ул. Митрополита Андрія Шептицького, 4, оф. 32..'

adress\_Intense\_2 = 'М. Контрактова площа вул. Хорива, 1-а'

adress\_Intense\_3 = 'М. ПЛОЩА ЛЬВА ТОЛСТОГО вул. Льва Толстого 9-а, 3 пов.'  
 Intense\_url\_1 = "https://cutt.ly/itDOJGo"  
 Intense\_url\_2 = "https://cutt.ly/4tDOZGg"  
 Intense\_url\_3 = "https://cutt.ly/7tDOVAp"

#### *# Графік роботи*

##### *# Київ*

Intense\_Work\_Schedule\_1\_0 = "Пн., Ср.: 10:00-20:00"  
 Intense\_Work\_Schedule\_1\_1 = "Вт., Чт.: 14:00-20:00"  
 Intense\_Work\_Schedule\_1\_2 = "Пт.: 10:00-20:00"

Intense\_Work\_Schedule\_2\_0 = "Пн., Ср.: 12:00-20:00"  
 Intense\_Work\_Schedule\_2\_1 = "Вт., Чт., Пт.: 10:00-20:00"

Intense\_Work\_Schedule\_3\_0 = "Пн. - Чт.: 10:00-20:00"  
 Intense\_Work\_Schedule\_3\_1 = "Пт.: 14:00-20:00"

#### *# Телефони для зв'язку*

##### *# Київ*

Intense\_Communication\_Phones\_1\_0 = "(099) 312-93-37"  
 Intense\_Communication\_Phones\_1\_1 = "(093) 170-27-42"  
 Intense\_Communication\_Phones\_1\_2 = "(044) 383-72-62"

Intense\_Communication\_Phones\_2\_0 = "(093) 170-27-40"  
 Intense\_Communication\_Phones\_2\_1 = "(063) 789-02-15"  
 Intense\_Communication\_Phones\_2\_2 = "(063) 233-10-06"  
 Intense\_Communication\_Phones\_2\_3 = "(044) 233-10-06"

Intense\_Communication\_Phones\_3\_0 = "(093) 170-27-31"  
 Intense\_Communication\_Phones\_3\_1 = "(093) 214-64-03"

#### *# Час заняття*

##### *# Київ*

Intense\_Choose\_time\_1\_0 = 'Пн. - Чт. на 08:10'  
 Intense\_Choose\_time\_1\_1 = 'Пн. - Чт. на 10:10'  
 Intense\_Choose\_time\_1\_2 = 'Пн. - Чт. на 12:00'  
 Intense\_Choose\_time\_1\_3 = 'Пн. - Чт. на 16:10'  
 Intense\_Choose\_time\_1\_4 = 'Пн. - Чт. на 18:10'  
 Intense\_Choose\_time\_1\_5 = 'Пн. - Чт. на 20:10'

Intense\_Choose\_time\_2\_0 = 'Пн. - Чт. на 08:10'  
 Intense\_Choose\_time\_2\_1 = 'Пн. - Чт. на 10:10'  
 Intense\_Choose\_time\_2\_2 = 'Пн. - Чт. на 12:00'  
 Intense\_Choose\_time\_2\_3 = 'Пн. - Чт. на 16:10'  
 Intense\_Choose\_time\_2\_4 = 'Пн. - Чт. на 18:10'  
 Intense\_Choose\_time\_2\_5 = 'Пн. - Чт. на 20:10'

Intense\_Choose\_time\_3\_0 = 'Пн. - Чт. на 08:10'  
 Intense\_Choose\_time\_3\_1 = 'Пн. - Чт. на 10:10'  
 Intense\_Choose\_time\_3\_2 = 'Пн. - Чт. на 12:00'  
 Intense\_Choose\_time\_3\_3 = 'Пн. - Чт. на 16:10'  
 Intense\_Choose\_time\_3\_4 = 'Пн. - Чт. на 18:10'  
 Intense\_Choose\_time\_3\_5 = 'Пн. - Чт. на 20:10'

#### *# Курс СТАНДАРТНИЙ*

Standard = ""

● **КУРС СТАНДАРТНЫЙ**

**4 МІСЯЦІ**

**СТАРТ: 7.09**

♥ **Ціна 5950 грн/семестр**

■ **3 зан. / нед. \* 1.5 ч.**

□ **2 Communication Classes + 1 Grammar Class**

**Графік: 8:10 / 9:10 / 9:50 / 10:50 / 15:00 / 16:40 / 18:20 / 20:00**

- 1. КОМФОРТНО:** цілий рівень за 4 місяці
  - 2. ЕФЕКТИВНО:** 3 заняття на тиждень по 1,5 години
  - 3. ІННОВАЦІЙНО:** навчання на базі унікальної гібридної платформи Notes by Green Forest
  - 4. ЗРУЧНО:** контроль за навчанням через додаток My Green Forest: домашні завдання, зміна графіка, успішність, новини, запис на додаткові заняття
  - 5. ПРОФЕСІЙНО:** всі викладачі Green Forest проходять підготовку в Green Forest Teaching Centre by Nadia Diachuk
  - 6. БЕЗКОШТОВНІ БОНУСИ:** онлайн версія підручника, розмовні клуби з носіями мови, спецкурси на різні граматичні та лексичні теми"
- price\_Standart = "5950 грн."

*# Київ*

address\_Standart\_1 = 'М. ЛІВОБЕРЕЖНА ул. Митрополита Андрія Шептицького, 4, оф. 32...'

address\_Standart\_2 = 'М. ПАЛАЦ СПОРТУ, вул. Рогнідинська, 4а.'

address\_Standart\_3 = 'М. ПОЗНЯКИ, ул. Гришка, ба.'

Standart\_url\_1 = "https://cutt.ly/itDOJGо"

Standart\_url\_2 = "https://cutt.ly/GtDJgdR"

Standart\_url\_3 = "https://cutt.ly/EtDJjCK"

*# Графік роботи*

*# Київ*

Standart\_Work\_Schedule\_1\_0 = "Пн., Ср.: 10:00-20:00"

Standart\_Work\_Schedule\_1\_1 = "Вт., Чт.: 14:00-20:00"

Standart\_Work\_Schedule\_1\_2 = "Пт.: 10:00-20:00"

Standart\_Work\_Schedule\_2\_0 = "Пн. - Пт.: 10:00-20:00"

Standart\_Work\_Schedule\_3\_0 = "Пн., Ср.: 10:00-20:00"

Standart\_Work\_Schedule\_3\_1 = "Вт., Чт., Пт.: 14:00-20:00"

*# Телефони для зв'язку*

*# Київ*

Standart\_Communication\_Phones\_1\_0 = "(099) 312-93-37"

Standart\_Communication\_Phones\_1\_1 = "(093) 170-27-42"

Standart\_Communication\_Phones\_1\_2 = "(044) 383-72-62"

Standart\_Communication\_Phones\_2\_0 = "(093) 170-27-38"

Standart\_Communication\_Phones\_2\_1 = "(093) 214-36-35"

Standart\_Communication\_Phones\_3\_0 = "(093) 170-27-31"

Standart\_Communication\_Phones\_3\_1 = "(093) 214-64-03"

**095) 672-26-52)"**

*# Час заняття*

*# Київ*

Standart\_Choose\_time\_1\_0 = '08:10'

Standart\_Choose\_time\_1\_1 = '09:10'

Standart\_Choose\_time\_1\_2 = '09:50'

Standart\_Choose\_time\_1\_3 = '10:50'

Standart\_Choose\_time\_1\_4 = '15:00'

Standart\_Choose\_time\_1\_5 = '16:40'

Standart\_Choose\_time\_1\_6 = '18:20'

Standart\_Choose\_time\_1\_7 = '20:00'

Standart\_Choose\_time\_2\_0 = '08:10'

Standart\_Choose\_time\_2\_1 = '09:10'

Standart\_Choose\_time\_2\_2 = '09:50'

Standart\_Choose\_time\_2\_3 = '10:50'

Standart\_Choose\_time\_2\_4 = '15:00'

Standart\_Choose\_time\_2\_5 = '16:40'

Standart\_Choose\_time\_2\_6 = '18:20'  
Standart\_Choose\_time\_2\_7 = '20:00'

Standart\_Choose\_time\_3\_0 = '08:10'  
Standart\_Choose\_time\_3\_1 = '09:10'  
Standart\_Choose\_time\_3\_2 = '09:50'  
Standart\_Choose\_time\_3\_3 = '10:50'  
Standart\_Choose\_time\_3\_4 = '15:00'  
Standart\_Choose\_time\_3\_5 = '16:40'  
Standart\_Choose\_time\_3\_6 = '18:20'  
Standart\_Choose\_time\_3\_7 = '20:00'

#### *# Реєстрація на курс*

Full\_Name\_Surname = "Введіть ПІБ"  
phone\_number\_text = "Введіть ваш контактний номер телефону"  
english\_test\_text = 'Пройдіть тест на визначення рівня англійської та виберіть Ваш рівень нижче'  
test\_text = "Пройти тест 📄"  
url\_test = "https://bit.ly/34VJJIn"  
english\_level\_text = "Тест на рівень знань англійської мови"  
registratio\_course\_1 = "Дякую! "+information+" ваш рівень англійської "+english\_skill+". Вас зареєстровано на курсу " +course+", час "+time\_lesson+ " за адресою "+ address+" .До сплати "+price+" Менеджер школи «Green Forest» зв'яжеться з Вами для визначення днів занять."  
pay = "Переглянути умови оплати 📄"  
url\_pay = "https://cutt.ly/YtOi7Hp"  
map\_school = "Школа на карті 📍"

#### *# Рівні англійської мови*

english\_level = "Оберіть ваш рівень"  
A1 = 'A1 - Beginner, Elementary '  
A2 = 'A2 - Pre-Intermediate '  
B1 = 'B1 - Intermediate '  
B2 = 'B2 - Upper-Intermediate '  
C1 = 'C1 - Advanced '

#### *# Замовити пробне заняття*

registratio\_course\_2 = "Дякую! " + information + " ваш рівень англійської " + english\_skill + ". Вас зареєстровано на пробне заняття курсу " + course + " за адресою " + address + ". Менеджер школи «Green Forest» зв'яжеться з Вами для визначення дати та часу."

#### *# Замовити дзвінок*

call\_text = 'Замовити дзвінок'  
comment\_yes = "Так ✓"  
comment\_no = "Ні ✗"  
add\_comment\_text = "Додати коментар?"  
write\_comment = "Введіть ваш коментарій"  
end\_call\_a = "Дякую! Працівники школи Green Forest зв'яжуться з Вами!"

payment = ""

Існують 3 способи оплати:

- повністю перед або після першого заняття. У цьому разі Ви отримуете знижку 10%.
  - першу половину одразу, другу - після другого заняття.
  - оплата частинами. Для цього Вам знадобиться кредитний ліміт на картці.
- Вартість книги не входить у вартість курсу. Книга коштує 450 грн.

#### *# Розклад розмовних клубів*

club\_text\_Kiev = "Розмовний клуб Київ 📍"  
club\_text\_Dnipro = "Розмовний клуб Дніпро 📍"  
club\_text\_Xarkiv = "Розмовний клуб Харків 📍"  
club\_text\_Odesa = "Розмовний клуб Одеса 📍"  
club\_text\_Lviv = "Розмовний клуб Львів 📍"

url\_Kiev = 't.me/english\_sunday\_greenforest'  
 url\_Dnipro = 't.me/englishsundaydnipro'  
 url\_Xarkiv = 't.me/englishsundaykh'  
 url\_Odesa = 't.me/englishsundayodessa '  
 url\_Lviv = 't.me/englishsundaylviv'

# Київ

where\_Kiev = ''

📍 **Where:**

Циферблат (вул. Володимирська, 49А). Клуб безкоштовний, проте окремо сплачується перебування в Циферблаті - 54 грн/год (у що вартість входять чай, кава, солодоці).

'''

when\_Kiev = ''

📅 **When:**

19:00 - every other Friday - Business English

12:00 - every Saturday - Speaking Club for Intermediate+ with native speaker

12:00 - every Sunday - English Sunday

'''

# Дніпро

where\_Dnipro = ''

📍 **Where:**

First Wave (вул. Короленко, 3).

'''

when\_Dnipro = ''

📅 **When:**

12:30 - every Saturday - Speaking Club for Intermediate+ with native speaker

11:00 - every Sunday - English Sunday

'''

# Харків

where\_Xarkiv = ''

📍 **Where:**

Green Forest Hub Kharkiv (вул. Дарвіна, 4)

'''

when\_Xarkiv = ''

📅 **When:**

19:30 - every other Friday - Business English

11:00 - every Saturday - Speaking Club for Intermediate+ with native speaker

13:00 - every Sunday - English Sunday

'''

# Одеса

where\_Odesa = ''

📍 **Where:**

Читальний зал бібліотеки ім. Грушевського (вул. Троїцька, 49/51).

'''

when\_Odesa = ''

📅 **When:**

19:00 - every other Friday - Business English

13:00 - every Sunday - English Sunday

'''

# Львів

where\_Lviv = ''

📍 **Where:**

MolodoZeleno (Джерельна, 20).

'''

when\_Lviv = ''

📅 **When:**

19:30 - every other Thursday - Business English

11:30 - every Saturday - Speaking Club for Intermediate+ with native speaker

12:00 - every Sunday - English Sunday

'''

# Найпоширеніші запитання



url\_questions = "https://telegra.ph/NAJPOSHIREN%D0%86SH%D0%86-ZAPITANNYA-04-15-2"  
# *Контакти*  
web\_cite = "Сайт школи ►"  
Facebook = "Facebook"  
Instagram = "Instagram"  
YouTube = "YouTube"  
Telegram = "Telegram"  
Pinterest = "Pinterest"  
CALL\_CENTER = "CALL CENTER ☎ "

url\_web\_cite = "greenforest.com.ua"  
url\_Facebook = "www.facebook.com/GreenForestSchool"  
url\_Instagram = "www.instagram.com/greenforest.english/"  
url\_YouTube = "www.youtube.com/channel/UC3iggK1DN7M3u5xpFDigpmw"  
url\_Telegram = "t.me/greenforestschool"  
url\_Pinterest = "www.pinterest.com/greenforest0662/"

CALL\_CENTER\_1 = "0 800 750 167"  
CALL\_CENTER\_2 = "(066) 920-10-10"  
CALL\_CENTER\_3 = "(068) 920-10-10"  
CALL\_CENTER\_4 = "(073) 920-10-10"

## ДОДАТОК Б. ІНТЕРФЕЙС ЧАТ-БОТА

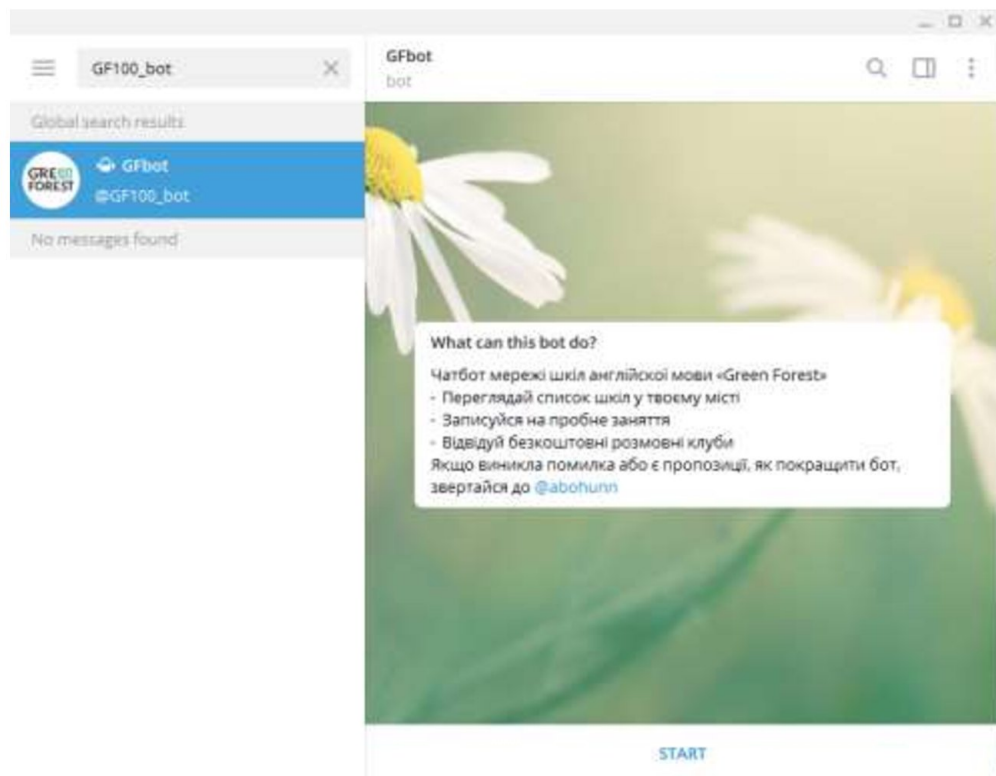


Рис.Б.1. Інтерфейс GFbot на Windows

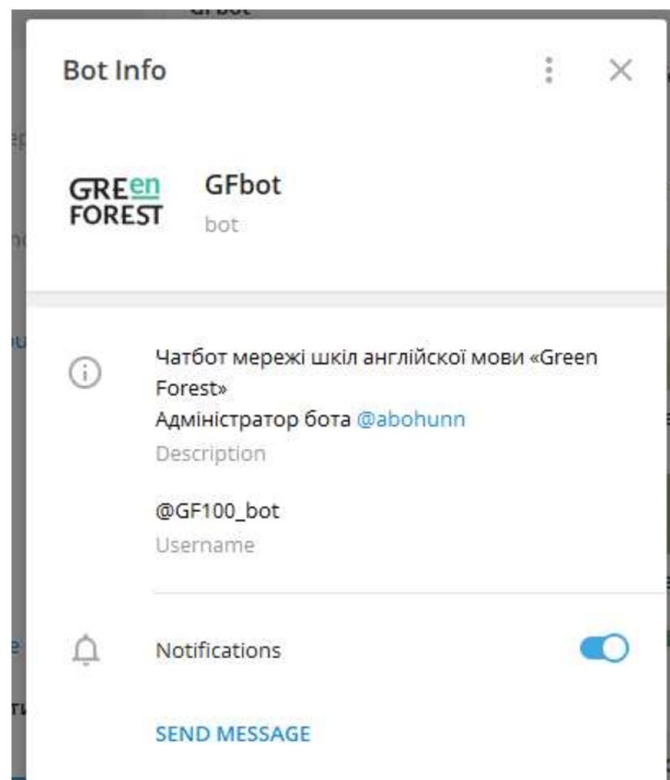
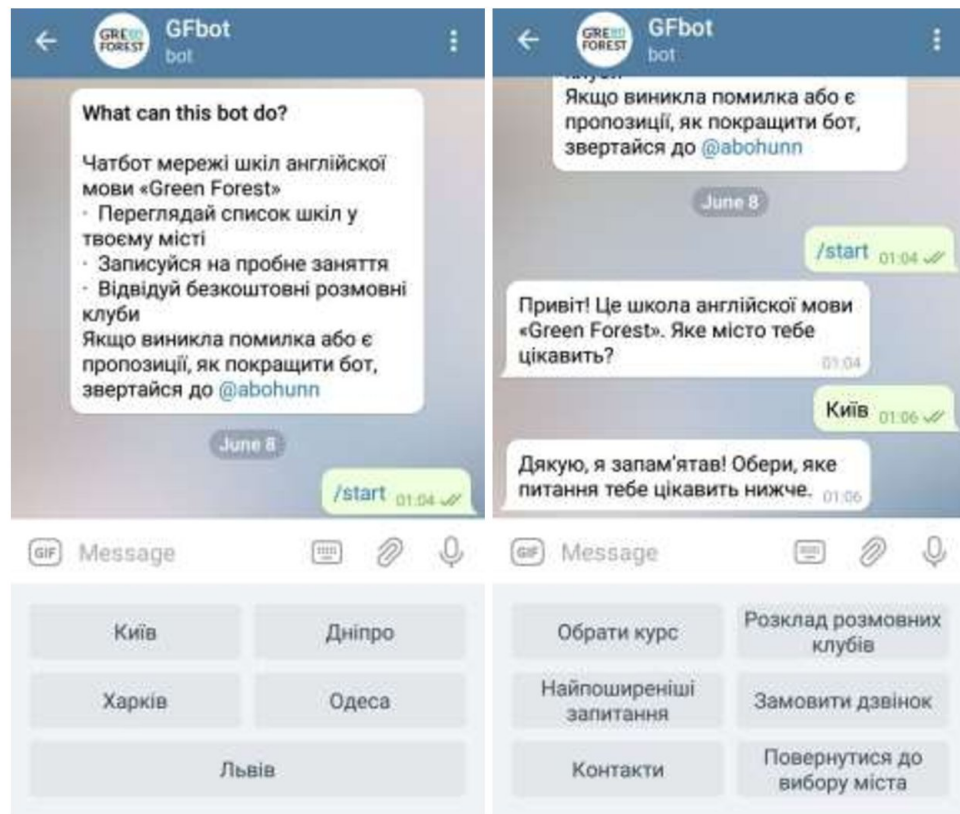


Рис.Б.2. Інформація про бота



а)

б)

Рис.Б.3. Інтерфейс GFbot на Android OS: а) меню вибору міста; б) головне меню



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
 ТЕХНОЛОГІЙ  
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## РОЗРОБКА CHAT-BOT ДЛЯ МЕРЕЖІ ШКІЛ ВИКЛАДАННЯ АНГЛІЙСЬКОЇ МОВИ ЗА ДОПОМОГОЮ PYTHON

Виконав студент 4 курсу  
 групи ПД-44  
 Шведченко Іван Сергійович  
 Керівник роботи  
 Коваленко Данило Сергійович

Київ – 2021

## МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

*Проговорити актуальність 2-3 речення після теми до мети*

- **Мета роботи:** отримати готову програму – Telegram-бот для мережі шкіл англійської мови «Green Forest» для зворотного зв'язку з можливістю запису інформації, введеної користувачем, у базу даних MySQL.
- **Об'єкт дослідження:** існуючі телеграм-боти
- **Предмет дослідження:** алгоритм створення чат-ботів для Telegram

## АНАЛОГИ



### Чат-бот «RailwayBot»

#### Переваги:

- можливість моніторингу та пошуку квитків цілодобово;
- розклад потягів завжди під рукою;
- функція повернення квитків.

#### Недоліки:

- квитки неможливо придбати через бот;
- доступні лише російська та українська мови.

### Чат-бот «NovaPoshta\_Bot»



#### Переваги:

- Відстеження відправлення не виходячи з меснджера;
- Інформативні повідомлення про статус посилки

#### Недоліки:

- Потребує вдосконалення безпеки особових даних користувача

3

## ТЕХНІЧНІ ЗАВДАННЯ

- проаналізувати обрану предметну область;
- порівняти існуючі аналоги чат-ботів;
- обрати технології та середовище для розробки;
- розробити та протестувати чат-бота на платформі Telegram.

4

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



TelegramBot  
API



5

## ЕТАПИ СТВОРЕННЯ ПРОЕКТУ

- Аналіз
- Проектування
- Програмування
- Тестування



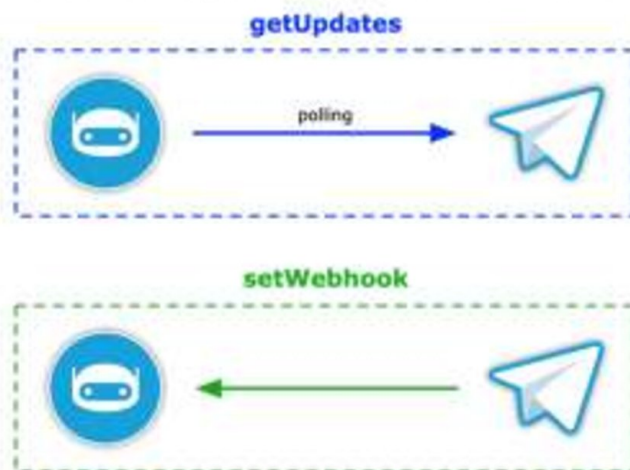
6

## ФУНКЦІЇ ЧАТ-БОТА

- інформування студентів;
- запис на курс/пробне заняття;
- проведення попереднього тестування;
- забезпечення зворотного зв'язку;
- надання миттєвих відповідей на найпоширеніші запитання.

7

### Вибір способу з'єднання із сервером



8

## Реєстрація бота у Telegram



9

## Результат розробки першого та головного меню



10

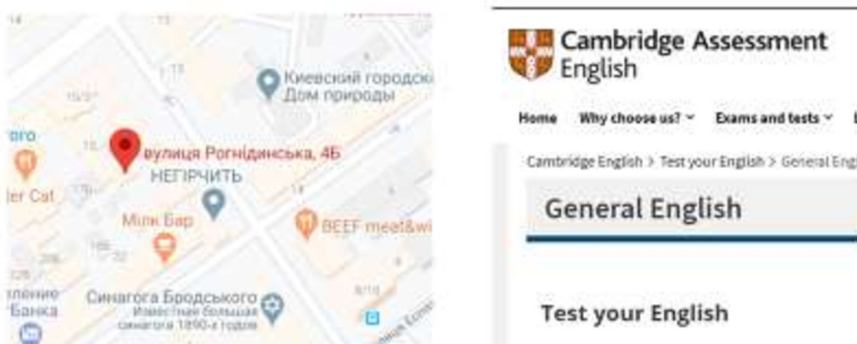


## Результат розробки вбудованих кнопок



11

## Додаткові можливості



12

## ВИСНОВКИ

- **Досягнуто мету:** розроблено Telegram-бот для мережі шкіл англійської мови «Green Forest» для зворотного зв'язку з можливістю запису інформації, введеної користувачем, у базу даних MySQL.
- Проаналізовано предметну область, визначено вимоги та функції помічника.
- Розроблено чат-бот та протестовано його швидкодію, працездатність на різних операційних системах, інтерфейс та функціональність.
- Проект має перспективу користуватись великим попитом у зв'язку з широкою доступністю, зручністю та зрозумілістю для пересічного користувача. У майбутньому планується вдосконалення та впровадження цілодобової роботи чат-бота за допомогою віддаленого сервера.

13