

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

Дипломний проект

здобувача освіти денної форми навчання

РП.06.02.000.ДП

***БОРИСКОВА
ДАНИЛА ЄВГЕНОВИЧА***

**м. Одеса
2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка клієнтської частини для веб-сайту ВСП «ОТФК ОНТУ»

Проектний матеріал складається з пояснювальної записки на 74 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах).

Дипломник Б. Іван (Боровецький Ю.Д.)

Керівник Шувалова І.О. (Шувалова І.О.)

Консультанти:

з економічної частини Копайгородська Т.Г. (Копайгородська Т.Г.)

з охорони праці Чорновол Н.І. (Чорновол Н.І.)

з дотримання вимог ЄСКД Петрашова В.І. (Петрашова В.І.)

старший консультант Кунуп Т.В. (Кунуп Т.В.)

До захисту допущений

Голова циклової комісії Кривченко Ю.В. (Кривченко Ю.В.)

Завідувач відділення Скорнякова О.В. (Скорнякова О.В.)

Захист « » 2023 р. Протокол ДКК №

Оцінка ДДК 5 (вирієнено)

Секретар ДДК

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР


Беркань І.В.

“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Боровецького Юрія Дмитровича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка клієнтської частини для веб-сайту ВСП «ОТФК ОНТУ»

затверджена наказом по коледжу від “ 17 ” жовтня 2023 р. № 235-А2-ОД

2. Термін здачі закінченого проекту (роботи) 09.06.2023

3. Вихідні данні до проекту (роботи)

1. Використання фреймворку React.js та Next.js для розробки клієнтської частини веб-сайту

2. Реалізація рендерингу на стороні сервера для розробки веб-сайту з використанням Next.js.

3. Використання мови програмування JavaScript та бібліотеки React.js для розробки клієнтської частини веб-сайту.

4. Реалізація логіки взаємодії елементів інтерфейсу та даних на клієнтській частині сайту

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Опис предметної області. Огляд бібліотек для клієнтської частини. Загальний технічний опис вирішення задачі дипломного проекту. Огляд інструментів та програмних рішень. Процес розробки дизайну та інтерфейсу веб-сайту. Проектування функціональності та взаємодії компонентів веб-сайту з використанням React.js. Реалізація функціональності та взаємодії клієнтської сторони з серверною частиною

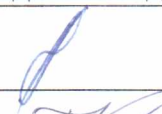
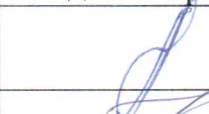

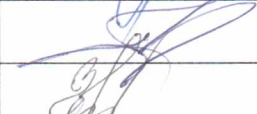
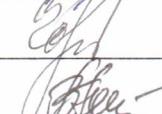
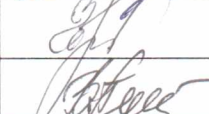

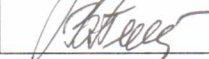
5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Титульний ліст, Огляд проекту, Плюси обраної архітектури, Написання клієнтської частини на React.js + Next.js,

Головна сторінка платформи, Написання мікросервісів на .NET та Python, Основні функціональні можливості,

Переваги веб-додатку для коледжу, Ризики та виклики, Підсумок

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний розділ	Шувалова І.О.		
Економічна частина	Копайгородська Т.Г.		
Охорона праці	Чорновол В.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання 01.05.2023

Керівник

Шувалова І.О.

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

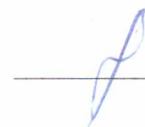
№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ	15.05.2023	Виконав
2.	Аналіз предметної галузі	16.05.2023	Виконав
3.	Огляд бібліотек для розробки клієнтської ч.	17.05.2023	Виконав
4.	Створення макету та інтерфейсу клієнтської частини	23.05.2023	Виконав
5.	Проектування логіки та взаємодії елементів веб-сайту	25.05.2023	Виконав
6.	Реалізація функціональності та взаємодії клієнтської сторони з серверною частиною	26.05.2023 28.05.2023	Виконав Виконав
7.	Тестування	31.05.2023	Виконав
8.	Робота над Охороною праці	04.06.2023	Виконав
9.	Робота над Економічною частиною	06.06.2023	Виконав
10.	Підготовка матеріалів до захисту	08.06.2023	Виконав

Дипломник



(підпис)

Керівник



(підпис)

Зміст

	стор.
ВСТУП.....	7
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	8
1.1 Аналіз предметної області. Мета і обґрунтування дипломного проекту	8
1.2 Актуальність обраної тематики дипломного проекту. Огляд та порівняння з існуючими рішеннями	9
1.2.1 Розбір аналогів проекту. Push-сповіщення у браузері	10
1.2.2 Прямі аналоги. Існуючі телеграм-боти для відстежування оновлень сучасної літератури.....	15
1.2.3 Огляд додатків для читання і відстежування	16
1.3 Вибір технологій розробки боту	19
1.3.1 Вибір мови програмування для реалізації проекту	20
1.3.2 Вибір середовища розробки для реалізації дипломного проекту.....	22
1.3.3 Вибір API та бібліотек для реалізації дипломного проекту.....	23
1.4. Розробка алгоритмів відстежування оновлень сучасної літератури.....	28
1.4.1 Опис функцій, які необхідно спроектувати для роботи з джерелами.....	31
1.4.2 Опис функцій, які необхідно спроектувати для роботи з базою даних.....	32
1.5 Опис етапів розробки проекту. Опис функціональності проекту....	33
1.5.1 Перший етап створення телеграм-боту, робота з BotFather	33
1.5.2 Завантаження Visual Studio Code та Python для початку роботи над кодом проекту	35
1.5.3 Структура кореневої папки проекту та встановлення бібліотек	37
1.5.4 Створення класів для відстежуваних витворів.....	40

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

1.5.5 Приклад функцій бота.....	41
2 ЕКОНОМІЧНА ЧАСТИНА.....	51
2.1 Резюме.....	51
2.2 Визначення трудомісткості розробки програмного забезпечення...	51
2.3 Розрахунок ціни програмного продукту.....	54
3 ОХОРОНА ПРАЦІ.....	56
3.1 Аналіз небезпечних та шкідливих чинників на робочому місці.....	57
3.2 Гігієнічні вимоги до виробничого середовища.....	58
3.3 Вимоги до організації робочого місця працівника.....	59
3.4 Пожежна безпека.....	60
ВИСНОВКИ.....	62
Перелік використаних джерел.....	64
ДОДАТОК А. Робота нескінченного циклу для відстежування оновлень сучасної літератури (мова Python).....	65
ДОДАТОК Б. Функції для додавання в базу даних нових творів та прив'язки творів до користувачів.....	68
ДОДАТОК В. Презентація.....	73

ВСТУП

Сучасні інформаційні технології та месенджери стали невід'ємною частиною нашого повсякденного життя. Вони забезпечують нам миттєвий зв'язок з іншими людьми та надають доступ до різних сервісів і контенту. Одним із таких популярних месенджерів є Telegram, який вирізняється своїми широкими можливостями для розробників.

У рамках цього дипломного проекту пропонується розробка телеграм-бота, спеціально призначеного для відстеження виходу нових глав сучасної літератури. Сучасна література упродовж десятиліть користується величезною популярністю і має мільйони шанувальників по всьому світу. Однак стежити за регулярними оновленнями і виходом нових глав може бути непросто, особливо з огляду на кількість випущених серій, величезну кількість ресурсів для читання, нерегулярність перекладу і велику кількість цих самих перекладачів.

Метою даного проекту є створення інтелектуального телеграм-бота, який автоматизує процес відстеження та надання інформації про вихід нових глав сучасної літератури. Бот буде інтегрований з платформою Telegram, що дасть змогу користувачам отримувати актуальні сповіщення та інформацію про вихід нових розділів на будь яких пристроях зі встановленим месенджером.

Основний акцент у цьому проекті буде зроблено на розробці ефективних алгоритмів збору та обробки даних, аналізі контенту, а також створенні зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. В якості основної мови програмування для реалізації проекту обрано Python, який є потужним і гнучким інструментом для розробки телеграм-ботів.

Цей дипломний проект є унікальною можливістю об'єднати в собі навички програмування, аналізу даних і розробки користувацького інтерфейсу. Він має великий потенціал для спрощення і поліпшення процесу отримання інформації про вихід нових глав сучасної літератури, що значно полегшить життя її фанатам і стане корисним інструментом для всіх, хто цікавиться цим популярним видом розваги.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Аналіз предметної області. Мета і обґрунтування дипломного проекту

Основна мета полягає у створенні чат-бота для платформи Telegram. Спочатку необхідно розробити привітальне повідомлення, а потім за допомогою програмного коду створити кнопки для забезпечення навігації користувача у системі. Перш за все, створюється п'ять основних кнопок: "Підписатися на оновлення літератури", "Відписатися від оновлень", "На що користувач підписан", "Вибір мови" та "Зв'язок з розробником".

Наступним кроком є створення інших меню з простішими кнопками, що вказують на цифру для подальшого вибору з варіантів після пошуку. Наприклад, при натисканні кнопки "Підписатися на оновлення літератури" користувачеві буде запропоновано вибрати жанр, а потім ввести назву твору. Після цього користувачеві пропонується вибрати слот для зберігання. При натисканні кнопки "Відписатися" користувачу відобразатиметься список всіх творів, на які він підписаний, і він матиме можливість відписатися від них. Функціонал кнопки "На що я підписаний" аналогічний, за винятком того, що відписатися неможливо.

Кнопка "Мова" дозволить користувачу змінювати мову інтерфейсу бота, а кнопка "Зв'язок з розробником" надаватиме посилання на Telegram-акаунт розробника. Всі літературні твори та дані користувачів зберігатимуться у базі даних, яка буде реалізована за допомогою системи управління базами даних MySQL.

Після завершення розробки бота, він зможе виконувати такі функції:

- Сповіщати користувача про вихід нових глав твору протягом 15 хвилин після публікації.
- Можливість відстежувати кілька творів одночасно.
- Швидкий пошук потрібного твору.
- Можливість відписуватися від оновлень твору.
- Можливість зв'язатися з розробником.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

- Можливість змінити мову інтерфейсу бота.

Основна ідея полягає у створенні зручного та корисного чат-бота, який спростить взаємодію користувача з літературними творами та надасть йому широкий функціонал.

1.2 Актуальність обраної тематики дипломного проекту. Огляд та порівняння з існуючими рішеннями

Читання японських коміксів та новел онлайн на пострадянському просторі і зокрема в Україні стало популярним і набуло значного поширення. Точну статистику читачів досить складно отримати, однак так як телеграм-бот займається відстежуванням виходу нових розділів саме на онлайн-ресурсах, деяку статистику все ж таки зібрати можна.

Наприклад, існує низка платформ і веб-сайтів, які надають японські комікси і новели мовами нашого регіону, і деякі з них публікують статистику про свою аудиторію і перегляди контенту. Наприклад, платформа Readmanga, одна з найпопулярніших на пострадянському просторі, повідомляє про мільйони унікальних відвідувачів і мільярди переглядів щомісяця. Це вказує на значний інтерес і активність користувачів у читанні манги онлайн.

Також варто зазначити, що форуми, групи в соціальних мережах і спільноти, присвячені коміксам та новелам, також слугують показниками їхньої популярності та активності. Наприклад, спільнота LiveLib, що спеціалізується на обговоренні літератури, включно з мангою та новелами, має мільйони зареєстрованих користувачів і тисячі активних обговорень.

Було проведено дослідження, для того, щоб знайти подібну статистику на якому-небудь ресурсі, який використовується безпосередньо в проекті. Структура проекту передбачає те, що запити від користувача спочатку обробляються API MangaDex, тому саме цей ресурс був взятий для отримання ще одних статистичних даних.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Цей ресурс на відміну від інших, які використовуються не обмежується якоюсь однією мовою, а підтримує всі існуючі у світі мови, тому статистика буде для усього світу. Сайт MangaDex на сторінці “Про нас” повідомляє наступну інформацію:

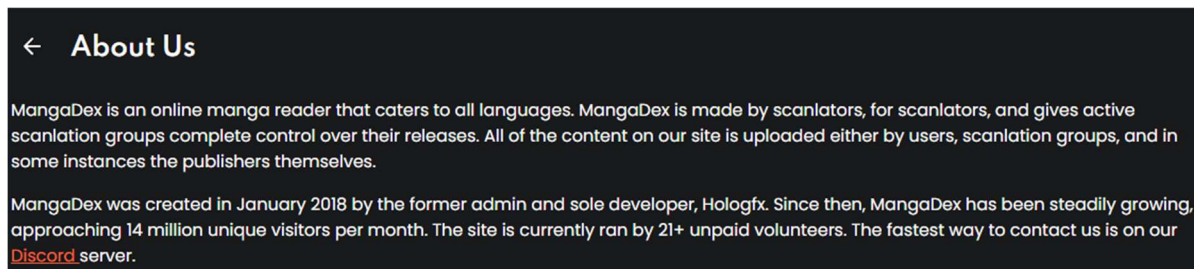


Рис. 1.1 - Сторінка “Про нас” на сайті MangaDex

Тобто на цьому сайті щомісяця є приблизно 14 мільйонів унікальних відвідувачів.

Таким чином, незважаючи на відсутність точних статистичних даних, доступні дослідження, дані платформ і активність в онлайн-спільнотах свідчать про значну кількість людей, які читають комікси і новели онлайн на пострадянському просторі і у всьому світі. Популярність цих форматів розваги продовжує зростати, а доступність онлайн-ресурсів і зручність читання за допомогою смартфонів і планшетів сприяють їхньому поширенню серед широкої аудиторії.

1.2.1 Розбір аналогів проекту. Push-сповіщення у браузері

Багато різних сайтів пропонують підписатися на витвір. Якщо це зробити, то користувачу будуть надходити сповіщення від браузера, коли новий розділ вийде. Послідовність дій, які потрібні для підписки дуже схожі на багатьох сайтах. У рамках дослідження, автор диплому спробував підписатись на який-небудь комікс на деяких ресурсах, котрі використовуються для проекту.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10



Рис. 1.2 - Кнопка “підписатися” на сторінці витвору на ресурсі mangapoisk.org

Після натискання кнопки “Підписатися”, у випадку з цим сайтом, користувачу пропонують увійти до свого аккаунта:

Рис. 1.3 - Сторінка входу на ресурсі mangapoisk.org

Була виконана авторизація використовуючи гугл аккаунт. Після чого, користувачу ще раз пропонують підписатися на нові розділи у сповіщенні:

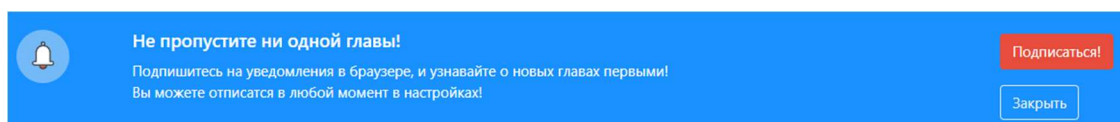


Рис. 1.4 - Сповіщення с пропозицією підписатися на вихід нових розділів

Браузер питає, чи дозволяється цьому ресурсу відправляти сповіщення:

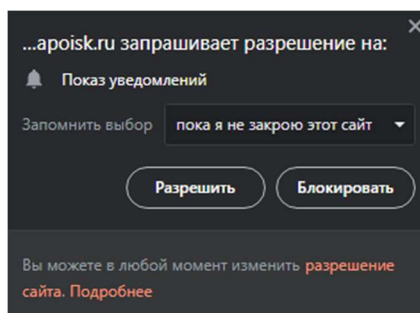


Рис. 1.5 - Запрос дозволи на показ сповіщень від сайту mangapoisk.com

Після дозволення уористувачу повинні будуть надходити push-сповіщення при виході нового розділу. Такий ж функціонал пропонує і інший ресурс, однак процес трохи складніший. Спочатку треба додати твір у розділ “читаю”:

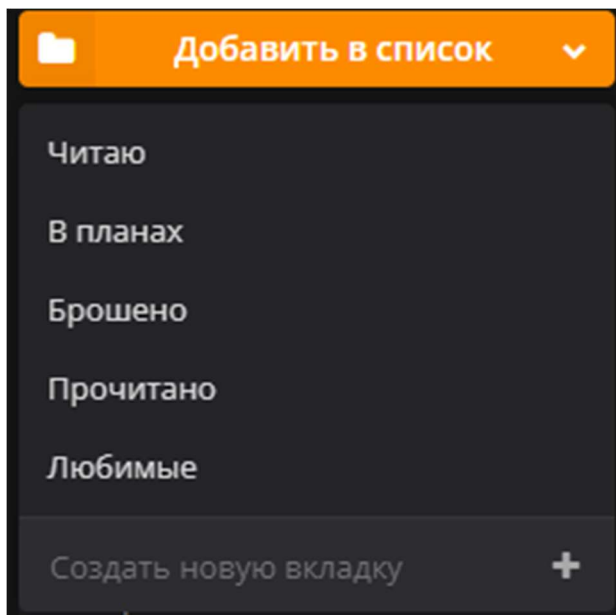


Рис. 1.6 - Процес додання твору до одного з списків на сайті mangalib.me

Далі треба зайти у налаштування свого аккаунту (на сайті обов’язкова реєстрація) і перейти у розділ “сповіщення” і налаштувати спочатку, з яких категорій користувач буде отримувати сповіщення:

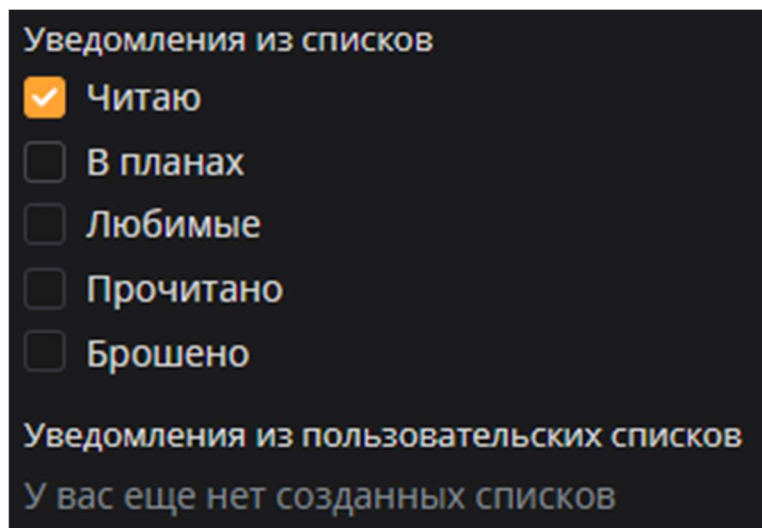


Рис 1.7 - Налаштування сповіщень у налаштуваннях аккаунту на сайті mangalib.me

А потім активувати функцію “надсилати push-сповіщення”:

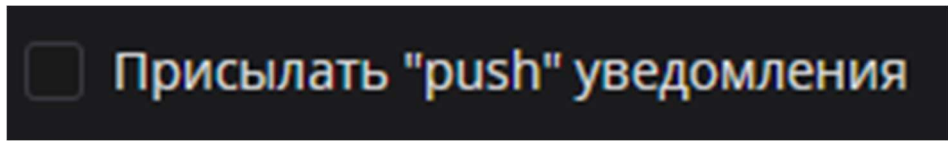


Рис. 1.8 - Кнопка “Надсилати “push” сповіщення” у налаштуваннях

Після чого браузер так само питає, чи дозволяти сайту надсилати ці самі сповіщення:

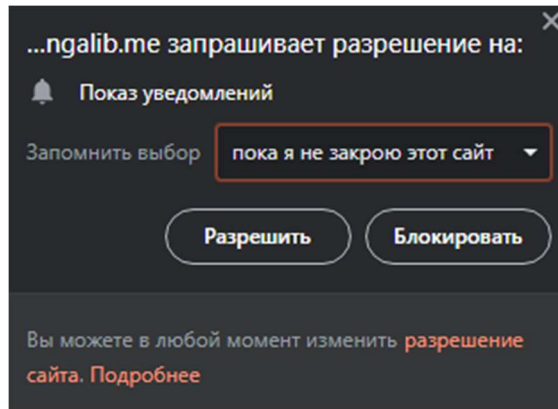


Рис. 1.9 - Запрос дозволу на показ сповіщень від сайту mangalib.me

Ну і наостанок було перевірено можливість відстежування нових розділів на основному ресурсі проекту - Mangadex.

Проходиться реєстрація:

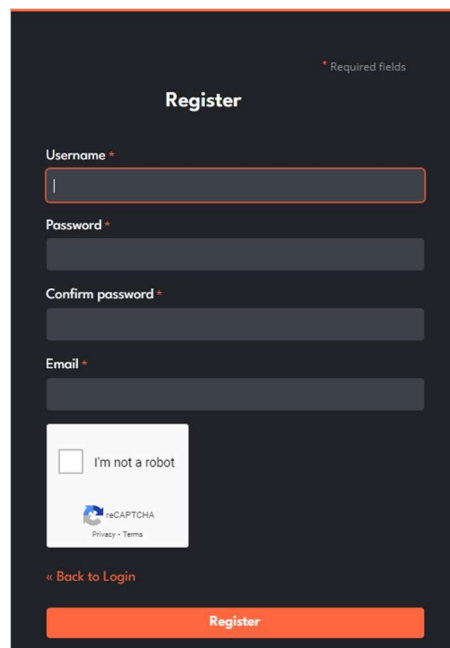


Рис. 1.10 - Форма реєстрації на сайті mangadex.org

Далі на сторінці потрібного твору натискається кнопка “Додати в бібліотеку”:

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

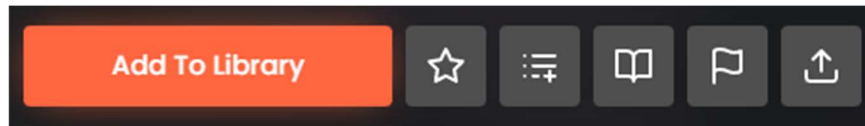


Рис. 1.11 - Кнопки взаємодії з твором на Mangadex

Після чого пропонується вибрати в який розділ додати твір:

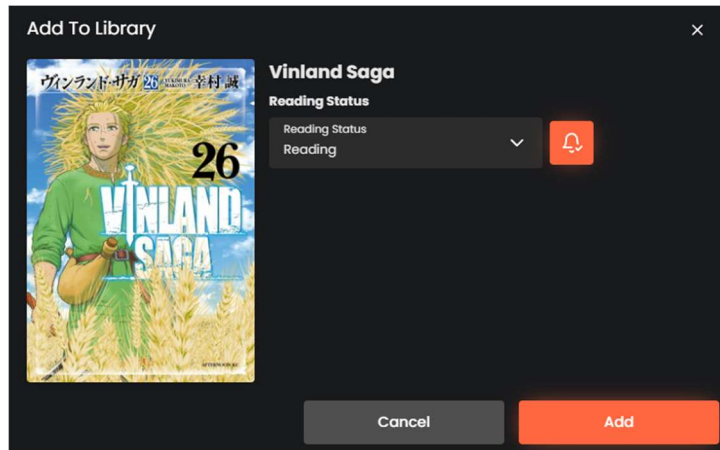


Рис. 1.12 - Меню додавання до списків

І після цього, наскільки зрозуміло, сайт не буде надсилати навіть push-сповіщень, а лише буде викладати нові глави у розділу “Оновлення” в кабінеті користувача:

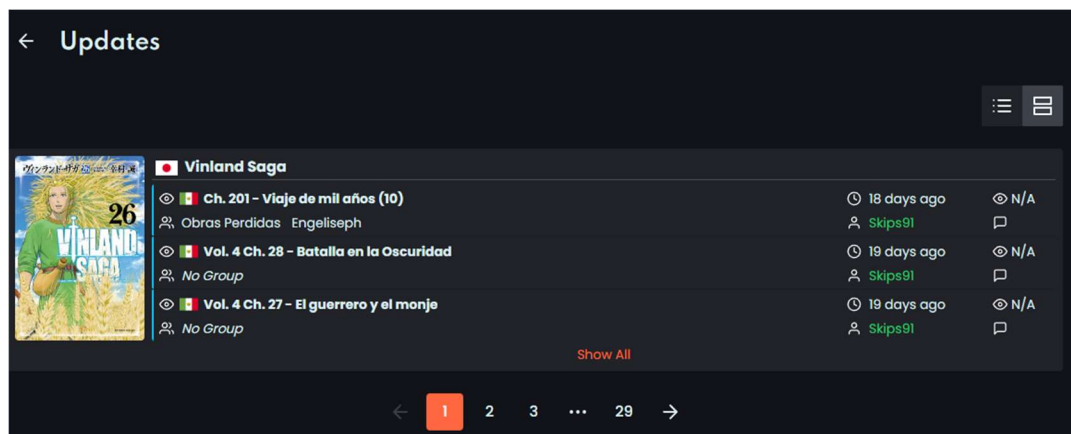


Рис 1.13 - Розділ “Оновлення” на Mangadex

Спосіб зі push-сповіщення має ряд недоліків у зрівнянні з концепцією розроблюваного у рамках дипломного проекту телеграм-боту, а саме:

- Необхідність реєстрації на сайтах для включення подібного функціоналу.

- Сповіщення які надсилає браузер можуть бути зручні не всім читачам.
- Прив'язка до пристрою.
- Сповіщення приходять тільки з одного сайту, а не з того, на якому новий розділ вийде швидше за інших.

1.2.2 Прямі аналоги. Існуючі телеграм-боти для відстежування оновлень сучасної літератури

На форумі одного з ресурсів, знайшлося таке обговорення:

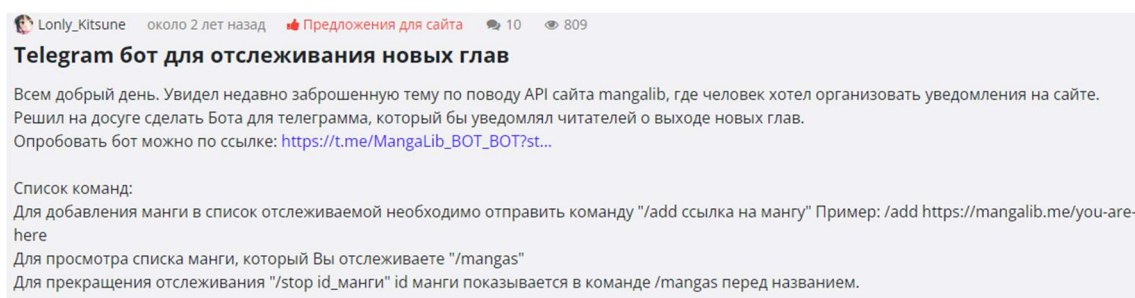


Рис. 1.14 - Обговорення на форумі mangalib про створення телеграм-боту для відстежування нових розділів.

В цілому, концепція цього бота така сама, як і у розроблюваного, однак є декілька відмінностей:

1. Бот перевіряє наявність нових розділів тільки на одному сайті, коли як бот з дипломного проекту буде перевіряти на багатьох.
2. Інтерфейс через команди, а не через кнопки, які більш зручні для користувача
3. Бот перевіряє наявність нових розділів тільки в одному сучасному літературному жанрі (азіатські комікси).
4. Бот підтримує лише російську мову.
5. Ну і найголовніший мінус - проект був розпочатий 2 роки тому і зараз не працює.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

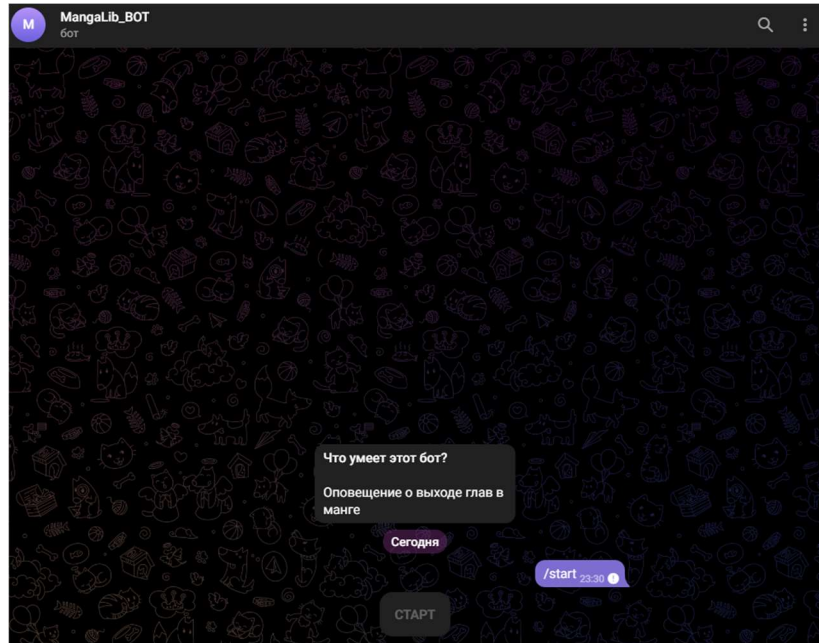


Рис. 1.15 - Демонстрація непрацездатності єдиного колись існуючого аналога мого боту

1.2.3 Огляд додатків для читання і відстежування

Проблема розроблюваного боту, що він працює лише для сповіщень, без можливості прямо у додатку прочитати твір, але по перше, телеграм-бот і не є додатком, який потрібно завантажувати, а по друге, це не зовсім так, тому-що у месенджері телеграм є вбудований браузер у якому і можна буде переглянути новий матеріал. Хоча це, можливо, не так зручно як у спеціальних додатках. Додатки, котрі одночасно є і так званими "читалками" для сучасної літератури. Саме такі додатки є найбільшими конкурентами для розроблюваного бота, якщо у них вбудована функція сповіщення, але мінусів в зрівнянні теж немало:

1. Прив'язка до одного додатку.
2. Прив'язка до одного пристрою.
3. Необхідність завантаження додатку.
4. Скоріш за все також треба буде пройти реєстрацію.

Однак є і плюси, наприклад необмежена кількість відстежуваних творів і можливість зручно прочитати розділ у додатку, а не переходити за посиланням на

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

сайт, до того-ж, іноді один розділ в боті можна буде прочитати на одному сайті, а наступний вже на іншому.

Для дослідження на смартфон був завантажений подібний додаток. Для пошуку такого, було введено в Google Play Market слово "ранобе":

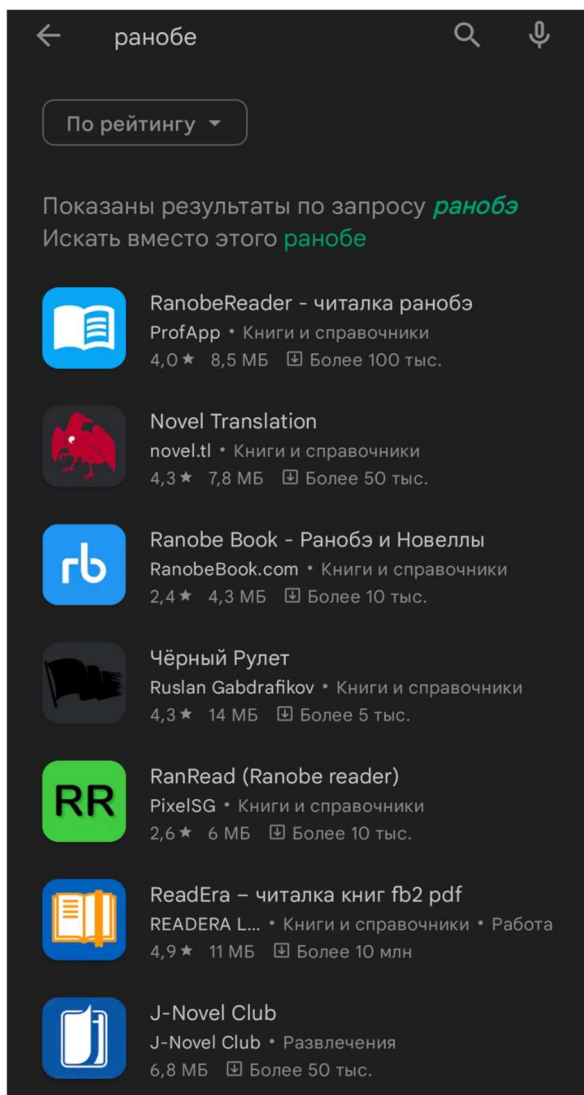


Рис. 1.16 - Результаты поиска у Play Market за словом “ранобе”

Перший ж результат, наскільки видно з опису підтримує відстежування:

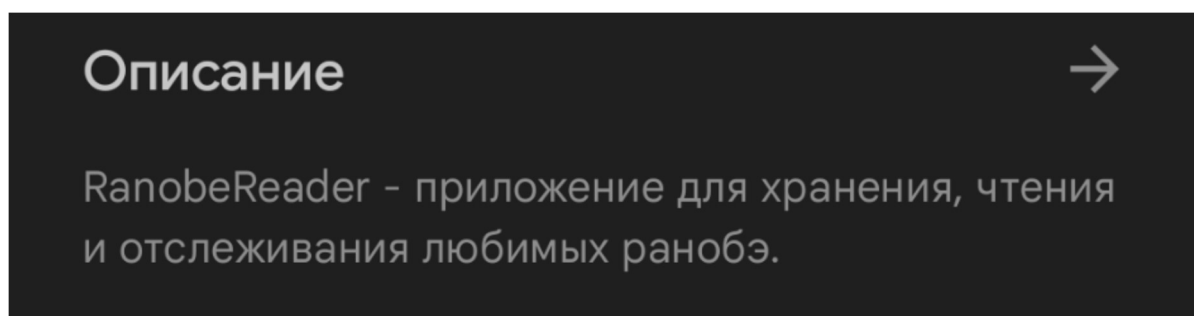


Рис. 1.17 - Опис додатку RanobeReader.

					РП 06. 02 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Також цей додаток, як і розроблюваний бот відстежує вихід нових розділів з багатьох різних ресурсів:

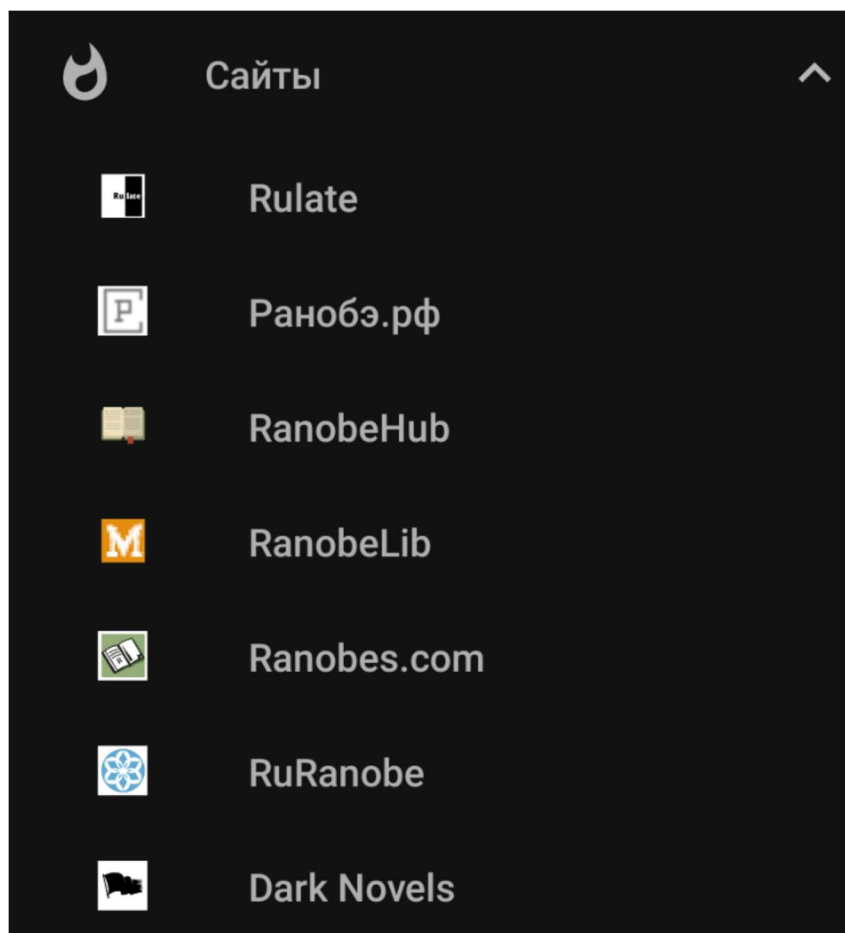


Рис. 1.18 - Демонстрація джерел у додатку

Серед однозначних мінусів - тільки російська мова, та відстежування тільки сучасної текстової літератури. Для відстежування наприклад коміксів треба завантажувати ще один додаток. Реєструватись не довелося, що є плюсом. Додаток має вбудований читач нових розділів і важить мало (трохи більше 8 мегабайт). Однак, це все ще сторонній додаток, коли як месенджер Telegram є у більшості людей. Також цей додаток є тільки для смартфонів, коли як месенджер Telegram можна встановити і на ПК, або взагалі користуватися веб версією.

Підсумовуючи можна сказати, що звісно ідея надсилати сповіщення при виході нового розділу будь якого жанру літератури не нова і це так чи інакше вже реалізовано. Однак усе питання у тому, наскільки це зручно. Розроблюваний бот буде надсилати сповіщення зручніше за всіх можливих конкурентів і охоплювати

майже усі жанри сучасної літератури на відміну від існуючих рішень, саме тому цей проект і є дуже актуальним.

1.3 Вибір технологій розробки боту

У даному підрозділі необхідно знайти найбільш підходящі мови програмування, АРІ та бібліотеки, для вдалого виконання дипломного проекту.

Кожна мова програмування має свої переваги, недоліки та сфери застосування. Вибір мови програмування залежить від вимог проекту, типу завдання, доступності ресурсів і досвіду розробника. Розробники можуть обирати між різними мовами залежно від своїх потреб та уподобань.

Так як розробляється телеграм-бот, був проведений невеликий аналіз, де звертаючись до інтернету було визначено на яких мовах найчастіше пишуть телеграм-ботів. Телеграм боти можуть бути написані різними мовами програмування, залежно від уподобань розробника. Телеграм надає АРІ для створення ботів, який підтримує кілька мов програмування. Деякі з популярних мов, які широко використовуються для створення телеграм ботів, включають:

- Python: Python є однією з найбільш популярних мов для створення телеграм-ботів. Для розробки телеграм-ботів на Python можна використовувати бібліотеку TeleBot.
- JavaScript: JavaScript є широко використовуваною мовою для створення веб-додатків і ботів. Для розробки телеграм-ботів на JavaScript можна використовувати бібліотеку node-telegram-bot-api.
- Java: Java також є популярною мовою програмування для розробки телеграм-ботів. Існують різні бібліотеки, такі як TelegramBots і TelegramBotsAPI, які полегшують створення ботів на Java.

Звісно, існує набагато більше мов програмування, на яких можна писати ботів, однак у більшості джерел, з яких бралася інформація щодо популярних мов для написання телеграм ботів ці три мови стояли у такому порядку, з чого можна

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

зробити висновок, що саме ці 3 мови є найпопулярнішими.

1.3.1 Вибір мови програмування для реалізації проекту

Для реалізації дипломного проекту була вибрана мова Python. Звертаючись до статистики, як от наприклад “TIOBE Index for May 2023”:

May 2023	May 2022	Change	Programming Language	Ratings	Change
1	1		Python	13.45%	+0.71%
2	2		C	13.35%	+1.76%
3	3		Java	12.22%	+1.22%
4	4		C++	11.96%	+3.13%
5	5		C#	7.43%	+1.04%
6	6		Visual Basic	3.84%	-2.02%
7	7		JavaScript	2.44%	+0.32%
8	10	▲	PHP	1.59%	+0.07%
9	9		SQL	1.48%	-0.39%
10	8	▼	Assembly language	1.20%	-0.72%

Рис. 1.19 - Перші 10 місць у рейтингу TIOBE

Або наприклад до “Github Language Stats”:

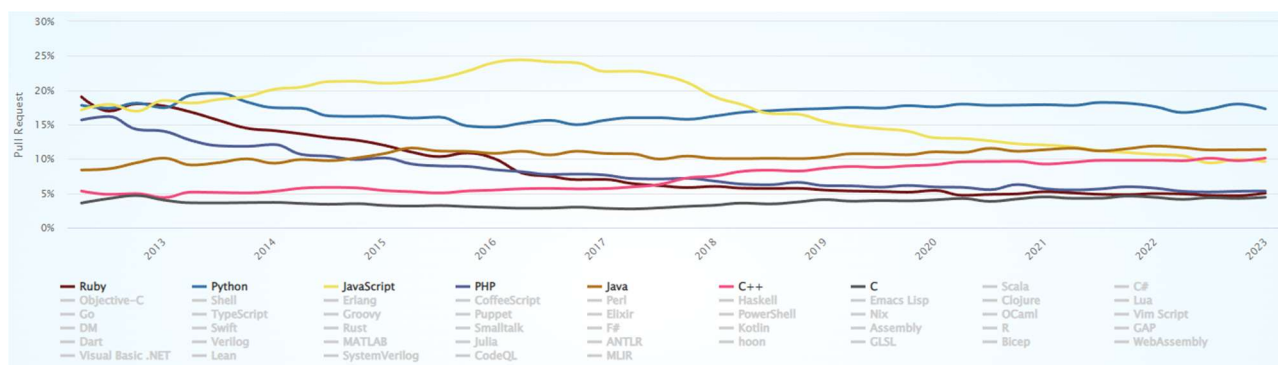


Рис. 1.20 - Популярність мов програмування на платформі GitHub

Можна зробити висновок, що Python зараз є найбільш популярною мовою програмування, а це значить, що з величезною кількістю проблем та помилок при розробці, з якою розробник може потенціально зустрітися, вже зустрічалися інші люди і ефективність рішення цих проблем буде вище, ніж в інших мовах. Серед інших причин вибору цієї мови можна назвати:

- Простота і зручність використання: Python має простий і лаконічний синтаксис, що робить його дуже читабельним і зрозумілим. Це дає змогу розробникам швидко писати і підтримувати код, що особливо корисно при створенні та підтримці телеграм-ботів.
- Велика кількість бібліотек і фреймворків: Python має велику екосистему бібліотек і фреймворків, які значно спрощують розробку телеграм-ботів. Наприклад, для роботи з Telegram API існують популярні бібліотеки, як-от python-telegram-bot і Telebot, які надають зручний інтерфейс для взаємодії з Telegram.
- Підтримка Telegram Bot API: Telegram надає потужний і гнучкий API для розробки ботів. Python має хорошу підтримку для роботи з Telegram Bot API, що дає змогу розробникам легко створювати й налаштовувати телеграм-ботів із використанням різних функцій, як-от надсилання повідомлень, опрацювання команд, робота з клавіатурами тощо.
- Розширюваність: Python має можливість інтеграції з іншими мовами програмування, такими як C/C++, що дає змогу використовувати продуктивні бібліотеки та модулі для оптимізації роботи телеграм-ботів. Це особливо корисно, коли потрібне виконання складних обчислень або обробка великих обсягів даних.
- Кросплатформеність: Python є кросплатформенною мовою, що означає, що код, написаний на Python, працюватиме на різних операційних системах, таких як Windows, macOS і Linux. Це дозволяє розробникам створювати телеграм-боти, які можуть працювати на будь-якій платформі.

Python надає розробникам потужні інструменти та простоту використання, що робить його прекрасним вибором для розробки телеграм-ботів. Завдяки своїй популярності та великій екосистемі, Python забезпечує розробникам необхідні ресурси та підтримку для створення ефективних і функціональних телеграм-ботів.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

1.3.2 Вибір середовища розробки для реалізації дипломного проекту

Для реалізації дипломного проекту було обрано середовище Visual Studio Code. Основні плюси VS Code включають:

- Легкість і швидке завантаження: VS Code має невеликий розмір і швидку швидкість роботи, що забезпечує комфортну розробку без затримок.
- Підтримка безлічі мов програмування: VS Code підтримує велику кількість мов програмування, включно з популярними, як-от JavaScript, Python, C++, Java, і багато інших. Це робить його підходящим для розробки дипломного проекту.
- Інтеграція з Git: VS Code має вбудовану підтримку системи контролю версій Git, що полегшує роботу з репозиторіями, комітами, порівнянням змін і управлінням гілками.
- Інструменти налагодження: VS Code надає зручні інструменти налагодження, що дають змогу розробникам перевіряти та виправляти помилки у своєму коді. Він підтримує налагодження для різних мов і дає змогу встановлювати точки зупинки, переглядати змінні та виконувати кроки налагодження.
- Розширюваність через плагіни: VS Code має широкі можливості для розширення функціональності за допомогою плагінів. Розробники можуть встановлювати різні розширення, що додають нові можливості, теми оформлення, інтеграцію з іншими інструментами та багато іншого.
- Активна спільнота: VS Code має велику й активну спільноту розробників, які діляться своїми напрацюваннями, створюють розширення і пропонують допомогу. Це дає змогу швидко знаходити розв'язання проблем, отримувати підтримку та обмінюватися досвідом з іншими розробниками.

Загалом, Visual Studio Code є зручним, легким і гнучким середовищем розробки, яке може бути застосоване для різних мов програмування і типів проектів. Його плюси включають простоту використання, підтримку Git,

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

налагоджувальні інструменти, розширюваність і підтримку активної спільноти розробників.

1.3.3 Вибір API та бібліотек для реалізації дипломного проекту

У проекті використовується безліч бібліотек, та 2 API.

Для роботи з TelegramBotAPI використовується бібліотека TeleBot. На те є декілька причин. Наприклад простота в використанні: TeleBot надає простий та інтуїтивно зрозумілий інтерфейс для взаємодії з Telegram Bot API. Завдяки цьому, почати роботу з бібліотекою і створення простих телеграм-ботів може бути досить легким і швидким процесом. Також бібліотека має великий функціонал. TeleBot пропонує широкий набір функцій, які дають змогу легко керувати і налаштовувати телеграм-бота. Бот зможе надсилати повідомлення, обробляти команди, працювати з клавіатурами, керувати файлами і медіа-ресурсами, а також багато іншого. Це дає змогу створювати більш складні та інтерактивні телеграм-боти з мінімальними зусиллями. Бібліотека підтримує оновлення Telegram API. TeleBot зазвичай активно оновлюється для підтримки останніх функцій і змін у Telegram Bot API. У бібліотеки є детальна документація та велика спільнота, яка може надати підтримку та допомогу в разі виникнення запитань чи проблем.

Ще одним API який використовується в проекті є MangaDex API, це інтерфейс програмування додатків, що надається MangaDex, популярною онлайн-спільнотою для читання манги. API надає розробникам доступ до різних функцій і даних, пов'язаних із мангою, які можна використовувати для створення застосунків, інтеграції з іншими сервісами або автоматизації певних завдань. За допомогою MangaDex API розробники можуть отримувати інформацію про мангу, таку як назви, описи, обкладинки, жанри, автори, глави, сторінки та багато іншого. Також доступні функції пошуку, що дають змогу шукати мангу за різними критеріями, наприклад, за назвою, жанром або автором. API також надає можливість отримувати інформацію про користувачів, коментарі та переклади. Використання MangaDex API дає змогу розробникам інтегрувати

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

функціональність MangaDex у свої додатки, створювати призначені для користувача інтерфейси для читання манги, створювати статистику або сповіщення про нові глави та випуски, а також проводити аналіз і обробку даних манги.

Для роботи з іншими ресурсами використовується парсинг, тобто процес вилучення даних з вихідного коду або структурованого тексту з метою отримання потрібної інформації. У контексті веб-розробки та даних мережі Інтернет, парсинг зазвичай відноситься до вилучення інформації з веб-сторінок або інших джерел даних. Парсинг виконується з використанням спеціальних програмних інструментів, які називаються парсерами, які аналізують структуру вихідного коду або тексту, розбиваючи його на більш дрібні елементи і витягуючи необхідні дані.

Для парсингу даних в проекті використовується бібліотека BeautifulSoup4. Це потужна і популярна бібліотека для вилучення даних з HTML і XML документів. Ось кілька причин, чому BeautifulSoup4 є хорошим вибором:

- Простота використання: BeautifulSoup4 надає простий і зрозумілий API, який дає змогу легко витягувати дані з HTML-коду. Він має інтуїтивний синтаксис, який робить процес парсингу більш доступним навіть для новачків.
- Гнучкість: BeautifulSoup4 може працювати з неправильним або складним HTML-кодом, автоматично виправляючи його і надаючи консистентну структуру даних для вилучення інформації. Він дає змогу обробляти різні типи тегів, атрибути та вміст, що робить його гнучким інструментом для різних сценаріїв парсингу.
- Підтримка Unicode: BeautifulSoup4 надійно обробляє різні символи Unicode, що особливо важливо під час роботи з багатомовними веб-сторінками або текстовими даними.
- Інтеграція з іншими бібліотеками: BeautifulSoup4 легко інтегрується з іншими бібліотеками Python, як-от Requests (для завантаження веб-сторінок) і Pandas (для аналізу даних). Це дає змогу створювати потужні скрипти для автоматизації збору та обробки даних.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Requests є ще однією бібліотекою, яка використовується в проєкті. Це популярна і проста у використанні бібліотека для надсилання HTTP-запитів у мові програмування Python. Вона надає зручні методи для роботи з різними типами запитів, підтримує HTTPS і SSL-сертифікати, дає змогу працювати із сесіями та кукісами, а також обробляє отримані HTTP-відповіді. Бібліотека Requests широко застосовується під час веб-скрапінгу, взаємодії з API та інших сценаріїв розробки.

Для того, щоб нескінченний цикл у проєкті не заважав боту опрацьовувати команди, треба цей цикл помістити у інший потік. Для багатопоточності використовується бібліотека `threading`. `threading` - це модуль у мові програмування Python, який дає змогу створювати та керувати потоками виконання в багатопотокових додатках. Потоки (`threads`) являють собою легковагі підзадачі, які можуть виконуватися паралельно всередині одного процесу, забезпечуючи багатозадачність.

Python забезпечує підтримку багатопоточності за допомогою модуля `threading`, який надає високорівневий інтерфейс для створення та керування потоками. Використання потоків може бути корисним у ситуаціях, коли кілька завдань мають виконуватися паралельно для поліпшення продуктивності або реагування на асинхронні події.

Основні поняття та функції, що надаються модулем `threading`, включають:

- Створення потоків: Модуль `threading` дає змогу створювати потоки шляхом створення підкласів від класу `Thread` і перевизначення методу `run()`, у якому визначається код, що виконується в потоці.
- Запуск потоків: Створені потоки можна запустити за допомогою методу `start()`, який запускає виконання методу `run()` в окремому потоці.
- Синхронізація потоків: Модуль `threading` надає різні примітиви синхронізації, як-от блокування (`Lock`), умови (`Condition`), семафори (`Semaphore`) тощо, що дають змогу керувати доступом до спільних ресурсів із різних потоків.
- Керування життєвим циклом потоків: Методи, як-от `join()` і `is_alive()`, дають змогу керувати життєвим циклом потоків, чекати на їхнє завершення і

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

перевіряти їхній статус.

- Робота з глобальними змінними: Під час роботи з кількома потоками необхідно враховувати синхронізацію доступу до глобальних змінних, щоб уникнути станів перегонів. Модуль threading надає відповідні механізми для безпечної роботи зі спільними даними.

Потоки можуть бути корисними для виконання завдань у фоновому режимі, обробки паралельних обчислень, очікування введення/виводу та інших сценаріїв, де потрібне паралельне виконання коду. Однак слід бути обережними під час роботи з потоками, оскільки неправильне використання може призвести до станів перегонів, блокувань та інших проблем синхронізації.

Для того, щоб нескінченний цикл не відправляв запити до бази даних та API постійно, а лише кожні 15 хвилин, я використовую бібліотеку time. Бібліотека time в Python надає функції та класи для роботи з часом і затримками. Вона дає змогу отримувати поточний час, вимірювати проміжки часу, задавати затримки у виконанні програми та багато іншого. Вона є частиною стандартної бібліотеки Python і широко використовується для керування часом і виконання затримок у програмі.

Для роботи з базами даних MySQL використовується додаток phpMyAdmin і бібліотека pymysql. MySQL - це система управління базами даних (СУБД), яка надає можливість зберігати, керувати та витягувати дані в структурованому форматі. Вона є однією з найпопулярніших СУБД і широко використовується веб-додатками та іншими системами для зберігання та обробки даних.

Я використовую саме цю СУБД для свого телеграм боту через:

- Широке поширення: MySQL є однією з найпопулярніших СУБД і має широку підтримку та використання в індустрії. Це означає, що існує велике співтовариство розробників і безліч ресурсів, як-от документація, підручники та форуми, які можуть допомогти в розробці та підтримці бази даних.
- Продуктивність і масштабованість: MySQL відомий своєю високою

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

продуктивністю і здатністю ефективно обробляти великі обсяги даних. Він також пропонує можливості масштабування, що дають змогу керувати зростанням бази даних у міру необхідності.

- Гнучкість і функціональність: MySQL підтримує широкий набір функціональних можливостей, включно з підтримкою різних типів даних, індексуванням, транзакціями, збереженими процедурами і тригерами. Це дає змогу розробникам ефективно моделювати та керувати даними, відповідаючи вимогам телеграм-бота.
- Сумісність з іншими технологіями: MySQL може легко інтегруватися з іншими технологіями та мовами програмування, що робить його гнучким вибором для розробки телеграм-бота. Його можна використовувати в поєднанні з різними веб-фреймворками, мовами програмування та іншими інструментами для створення повнофункціонального і надійного бота.
- Надійність і безпека: MySQL має механізми гарантування безпеки даних, включно з можливістю шифрування, авторизацією та управлінням доступом. Він також відомий своєю стабільністю і надійністю, що є важливим аспектом під час розроблення та обслуговування телеграм-бота.

phpMyAdmin - це веб-додаток, призначений для керування базами даних MySQL через веб-інтерфейс. Він написаний мовою програмування PHP і надає зручний та інтуїтивно зрозумілий інтерфейс для виконання різноманітних операцій із базами даних, як-от створення, видалення та зміна таблиць, вставлення, оновлення та видалення даних, виконання SQL-запитів, керування користувачами та привілеями і багато іншого. phpMyAdmin дає змогу розробникам і адміністраторам легко адмініструвати бази даних MySQL без необхідності використання командного рядка або інших інструментів керування.

Бібліотека pymysql - це бібліотека для мови програмування Python, яка надає простий спосіб взаємодії з базами даних MySQL. Вона дає змогу виконувати різні операції з базою даних, як-от виконання запитів, створення таблиць, вставлення, оновлення та видалення даних. Pymysql надає зручний та інтуїтивно зрозумілий

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

інтерфейс для роботи з MySQL. З її допомогою ви можете встановити з'єднання з базою даних, виконати SQL-запити та отримати результати. Бібліотека забезпечує безпеку і захист від атак SQL-ін'єкцій, надаючи механізми параметризованих запитів. Основні можливості бібліотеки pymysql включають:

- Встановлення з'єднання з базою даних MySQL.
- Виконання SQL-запитів і отримання результатів.
- Підтримка параметризованих запитів для безпечного опрацювання користувацьких даних.
- Робота з транзакціями для забезпечення цілісності даних.
- Створення, зміна та видалення таблиць і баз даних.
- Управління користувачами та привілеями бази даних.
- Обробка помилок і винятків під час роботи з базою даних.

1.4 Розробка алгоритмів відстежування оновлень сучасної літератури

У теорії, для відстежування оновлень підходить алгоритм такого типу:

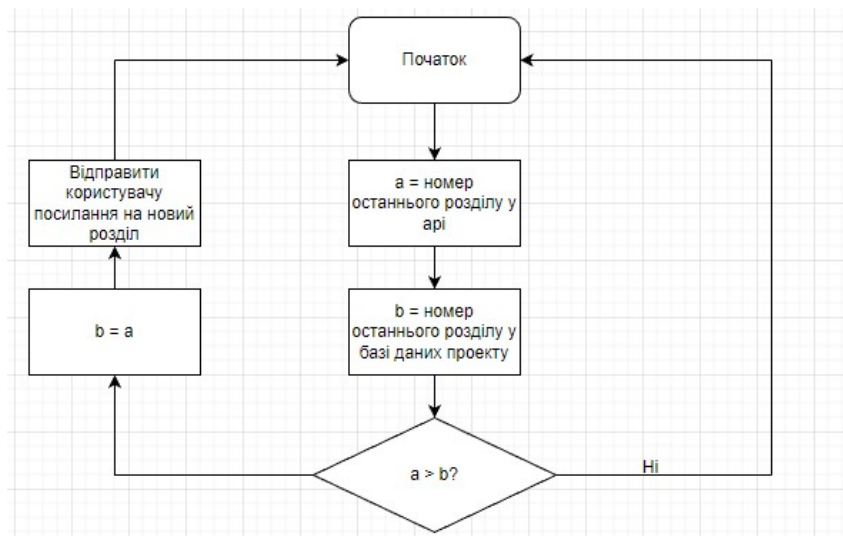


Рис. 1.21 - Загальний алгоритм для відстежування оновлень

Для відстежування оновлень для новел цей алгоритм і був примінен, бо для актуалізації даних у цьому літературному жанрі використовується 1 джерело. Для відстежування оновлень коміксів, до алгоритму, представленого вище, треба

внести деякі правки, бо цей жанр літератури відстежується не з одного джерела, тому оновлений алгоритм для коміксів виглядає так:

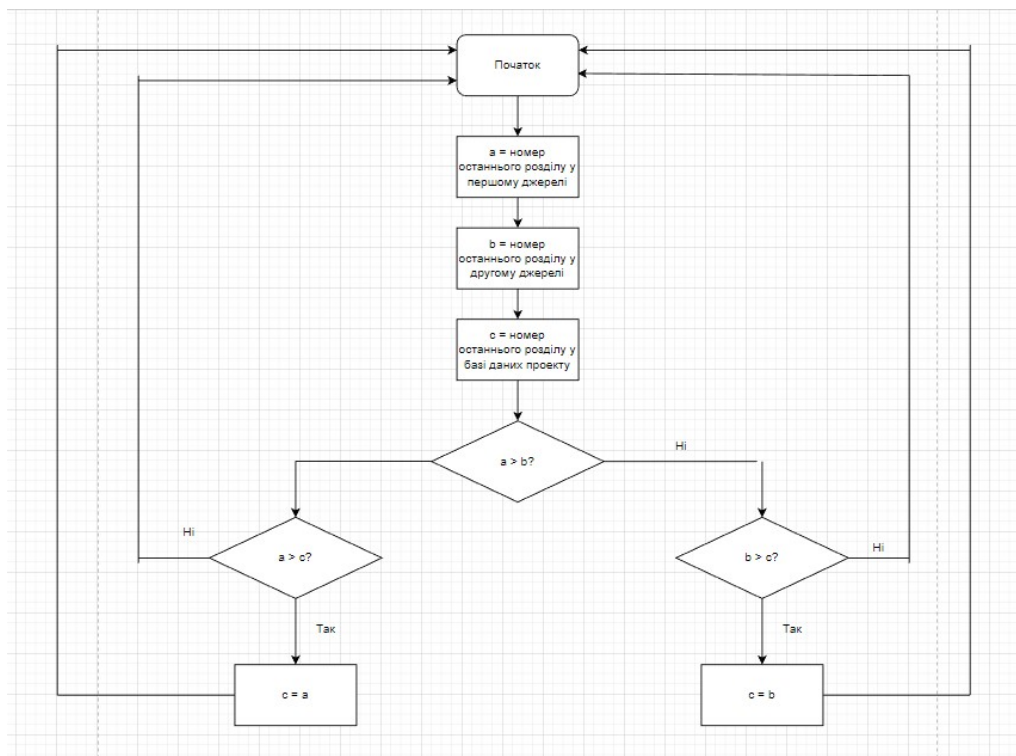


Рис. 1.22 – Алгоритм для відстежування оновлень коміксів у проекті

Число змінних a і b представлених на блок схемах бот буде отримувати за допомогою запитів до API, чи за допомогою парсингу, як-от наприклад:

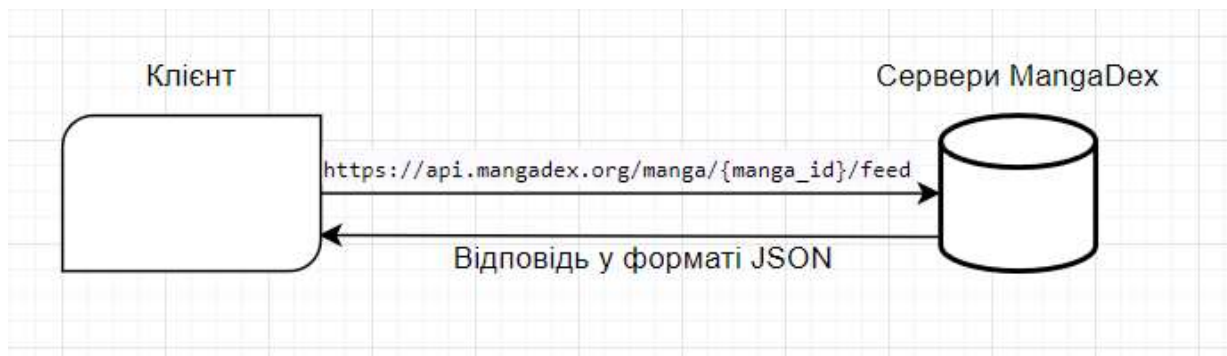


Рис. 1.23 – Схема запиту до MangaDex API

Це найбільш примітивний опис алгоритму отримання даних щодо останнього розділу, який є на серверах MangaDex. Однак на ділі він є більш складним, так як необхідно вилучити конкретні дані. Почати треба з того, що GET запит feed повертає інформацію про усі існуючі розділи на усіх мовах перекладу від усіх

Зм.	Арк.	№ докум.	Підпис	Дата

перекладачів. Просто так отримати інформацію про останній розділ не вийде ще й тому, що дані видаються лише по 500 розділів на відповідь, тобто якщо витвір має наприклад 200 глав і перекладений на 7 мов, то загальна кількість розділів буде складати 1400. Виникає потреба у розробці алгоритму для сортування усіх отриманих результатів. Для цього необхідно виконати такі дії:

1. Проініціалізувати масиви `last_chapter_ua`, `last_chapter_ru` і `last_chapter_en` з початковими значеннями `['none', 0.0, 'none']` для зберігання інформації про останні розділи різними мовами (українською, російською та англійською відповідно).
2. Сформуванати URL для запиту до API MangaDex, використовуючи переданий ідентифікатор манги. Цей запит представлений у рисунку 1.23
3. Відправити GET-запит за отриманим URL і отримати відповідь.
4. Якщо код стану відповіді дорівнює 200 (успішний запит), то функція обробляє дані з відповіді.
5. Витягти загальну кількість глав манги з відповіді та проініціалізувати змінні `offset` (зміщення) і `manga_chapters` (список глав манги).
6. У циклі виконати послідовні запити з використанням різних значень зсуву (`offset`) для отримання всіх розділів манги і додати їх у список `manga_chapters`.
7. Перебирати кожен главу манги у списку `manga_chapters` і порівняти її мову з цільовими мовами ('uk', 'ru', 'en').
8. Якщо розділ відповідає мові та його номер (у вигляді числа) більший, ніж поточний номер останнього розділу відповідною мовою, то оновити інформацію про останній розділ для цієї мови.
9. Повернути список з інформацією про останні розділи манги різними мовами в такому форматі: `[last_chapter_en, last_chapter_ru, last_chapter_ua]`, тобто повернути двувимірний масив.
10. Якщо код стану відповіді не дорівнює 200, то функція повинна вивести повідомлення про помилку і повернути `None`.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

1.4.1 Опис функцій, які необхідно спроектувати для роботи з джерелами

- `get_manga_id(title)`: Ця функція буде приймати назву манги (`title`) і повертати ідентифікатори (ID) відповідних манг на основі запиту до API MangaDex.
- `get_manga_names(manga_id)`: Функція буде приймати ідентифікатор манги (`manga_id`) і повертати список з іменами манги різними мовами (японською, російською та англійською) на основі запиту до API MangaDex.
- `get_image_link(manga_id)`: Функція буде приймати ідентифікатор манги (`manga_id`) і повертати пряме посилання на обкладинку манги на основі запитів до API MangaDex.
- `get_manga_chapters(manga_id)`: Ця функція прийматиме ідентифікатор манги (`manga_id`) і повертатиме інформацію про останні глави манги різними мовами (англійською, російською та українською) на основі запитів до API MangaDex.
- `is_not_completed(manga_id)`: Функція прийматиме ідентифікатор манги (`manga_id`) і перевірятиме, чи є манга завершеною (`completed`) на основі запиту до API MangaDex.
- `mangapoisk_last_chapter(name)`: Ця функція приймає назву манги (`name`) і повертає інформацію про останній розділ манги на основі парсингу сайту "mangapoisk.org".
- `novela_all_data(name)`: Функція прийматиме назву новели (`name`) і повертатиме різні дані про них, включно з номером останнього розділу, посиланням на останній розділ, назвою, посиланням на твір, посиланням на обкладинку, описом та інформацією про завершеність твору на основі парсингу сайту "ranobe.me".
- `novela_data_id(id)`: Функція приймає ідентифікатор новели (`id`) і повертає різні дані про неї, включно з номером останнього розділу, посиланням на останній розділ, назвою на основі запиту до сайту "ranobe.me".

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

1.4.2 Опис функцій, які необхідно спроектувати для роботи з базою даних

- Функція `add_to_database_manga` використовується для додавання інформації про мангу в базу даних. Вона приймає такі параметри:
`manga_id`: ідентифікатор манги;
`manga_name`: назва манги;
`last_chapter_ru_number`: номер останнього розділу російською мовою;
`last_chapter_ru_link`: посилання на останній розділ російською мовою;
`last_chapter_ua_number`: номер останнього розділу українською мовою;
`last_chapter_ua_link`: посилання на останній розділ українською мовою;
`last_chapter_en_number`: номер останнього розділу англійською мовою;
`last_chapter_en_link`: посилання на останній розділ англійською мовою;
`last_chapter_mangapoisk_number`: номер останнього розділу на Mangapoisk;
`last_chapter_mangapoisk_link`: посилання на останній розділ на Mangapoisk.
Функція перевіряє, чи є вже манга з таким ідентифікатором у базі даних. Якщо манга вже є, нічого не робить. Якщо манги немає, то вона додає запис про мангу в базу даних.
- Функція `add_to_database_user` використовується для додавання інформації про користувача в базу даних. Вона приймає параметр `chat_id`, який є ідентифікатором користувача. Функція перевіряє, чи є вже користувач із таким ідентифікатором у базі даних. Якщо користувач уже є, нічого не робить. Якщо користувача немає, то вона додає запис про користувача в базу даних.
- Функція `delete_art_from_user` використовується для видалення інформації про твір (мангу або новелу) у користувача. Вона приймає параметри `user_id` - ідентифікатор користувача та `slot_number` - номер слота, в якому знаходиться твір.
- Функція `list_of_subscribes` використовується для отримання списку підписок користувача. Вона приймає параметр `chat_id` - ідентифікатор користувача і повертає список слотів підписок, у яких містяться ідентифікатори творів.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

- Функція `add_manga_to_user` використовується для додавання манги в слот підписки користувача. Вона приймає параметри `manga_id` - ідентифікатор манги, `user_id` - ідентифікатор користувача та `slot_number` - номер слота, у який потрібно додати мангу.
- Функція `add_to_database_novel` використовується для додавання інформації про новелу в базу даних. Вона приймає такі параметри:
`last_chapter_number`: номер останнього розділу новели;
`last_chapter_link`: посилання на останній розділ новели;
`novel_name`: назва новели;
`novel_link`: посилання на новелу.
- Функція `add_novel_to_user` використовується для додавання новели в слот підписки користувача. Вона приймає параметри `novel_id` - ідентифікатор новели та `user_id` - ідентифікатор користувача.

1.5 Опис етапів розробки проекту. Опис функціональності проекту

1.5.1 Перший етап створення телеграм-боту, робота з BotFather

Для створення ботів в месенджері телеграм, спочатку треба скористатися іншим ботом, так званим BotFather.

BotFather - це офіційний бот-асистент, що надається Telegram, який допомагає користувачам створювати та керувати своїми власними ботами в месенджері Telegram. Він є ключовим інструментом для розробників, які хочуть створити і налаштувати бота.

Для створення нового бота необхідно виконати команду `/newbot` у `@BotFather`. Під час виконання команди `@BotFather` запитає у вас ім'я та логін (ім'я користувача) для нового бота. Потім він згенерує унікальний токен авторизації для вашого бота.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

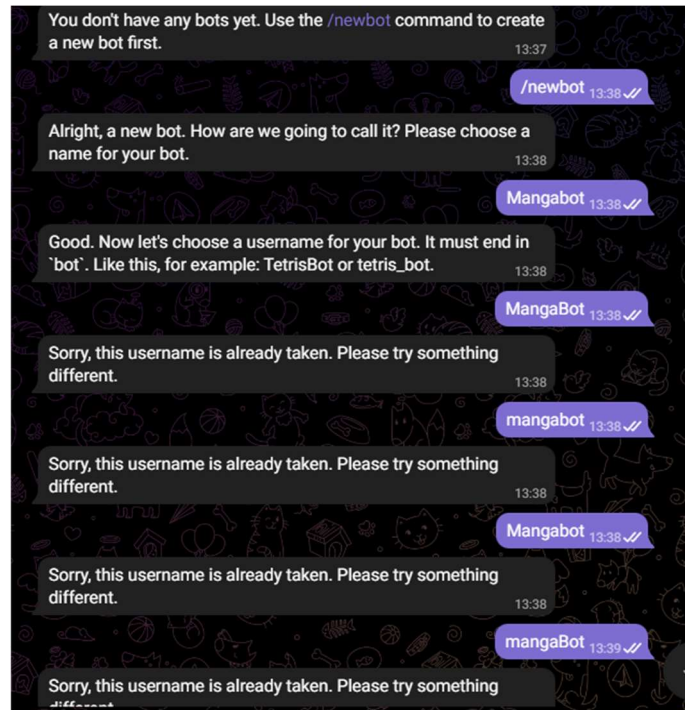


Рис. 1.24 - Демонстрація старту роботи з BotFather і перебор імен для бота
ДИПЛОМНОГО ПРОЄКТУ

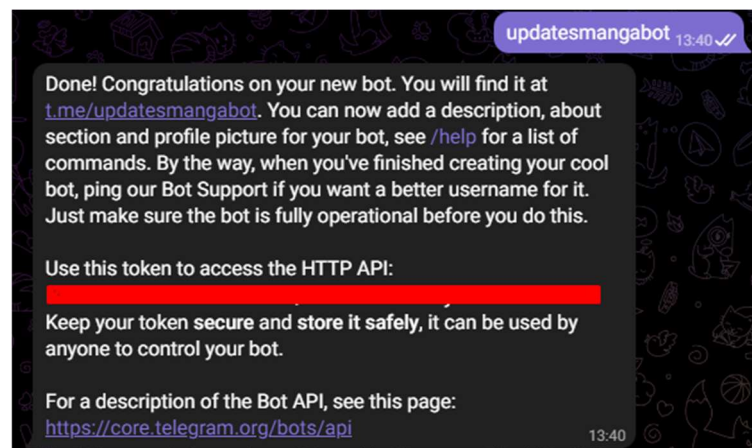


Рис 1.25 - Після підбору назви, бот надає токен

Токен являє собою унікальний рядок виду 228123456:QWerTyUIOPASdfGhJKLzXcvbNMQ-AB1CDE. Він необхідний для авторизації бота і надсилання запитів через API бота.

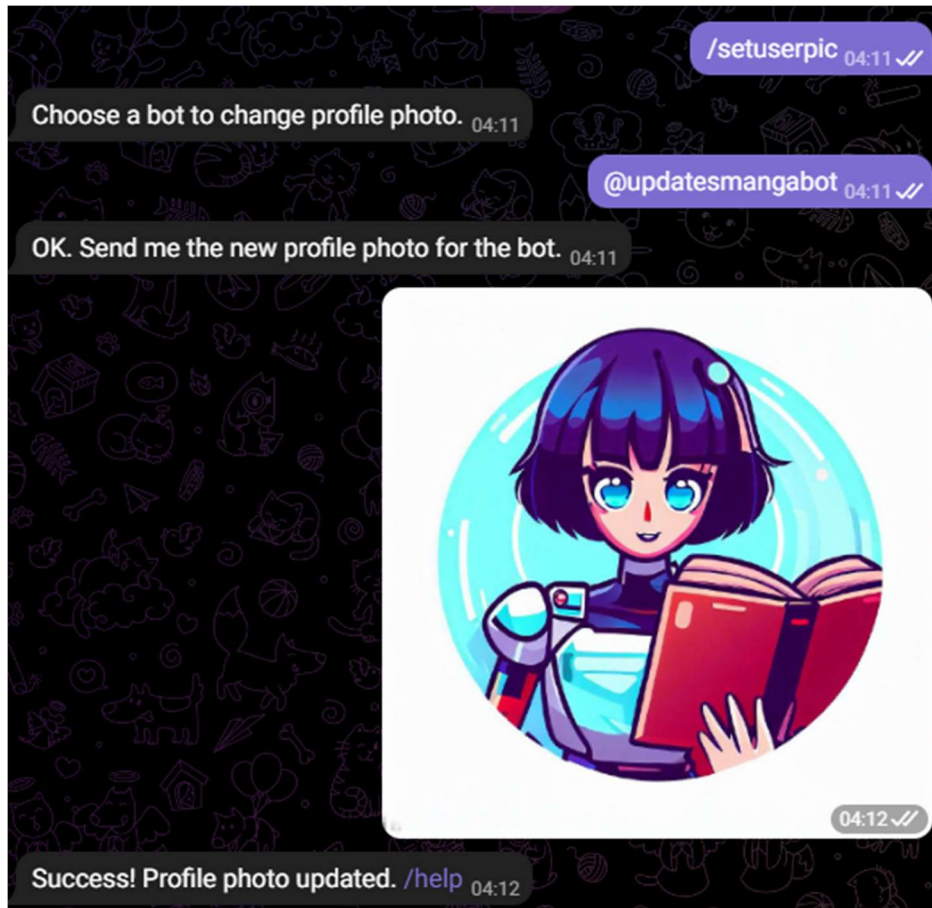


Рис. 1.26 - Демонстрація команди /setuserpic в BotFather. Дозволяє змінити аватарку бота.

1.5.2 Завантаження Visual Studio Code та Python для початку роботи над кодом проекту

Для того, щоб почати роботу з Python, треба його встановити. Встановлення Python на комп'ютер може незначно відрізнятись залежно від операційної системи. Після невеликого аналізу вдалось сформуванати докладні інструкції для Windows по встановленню Python та VS Code, бо саме на цій операційній системі і розробляється проект:

1. Необхідно перейти на офіційний сайт Python за адресою <https://www.python.org/downloads/>.

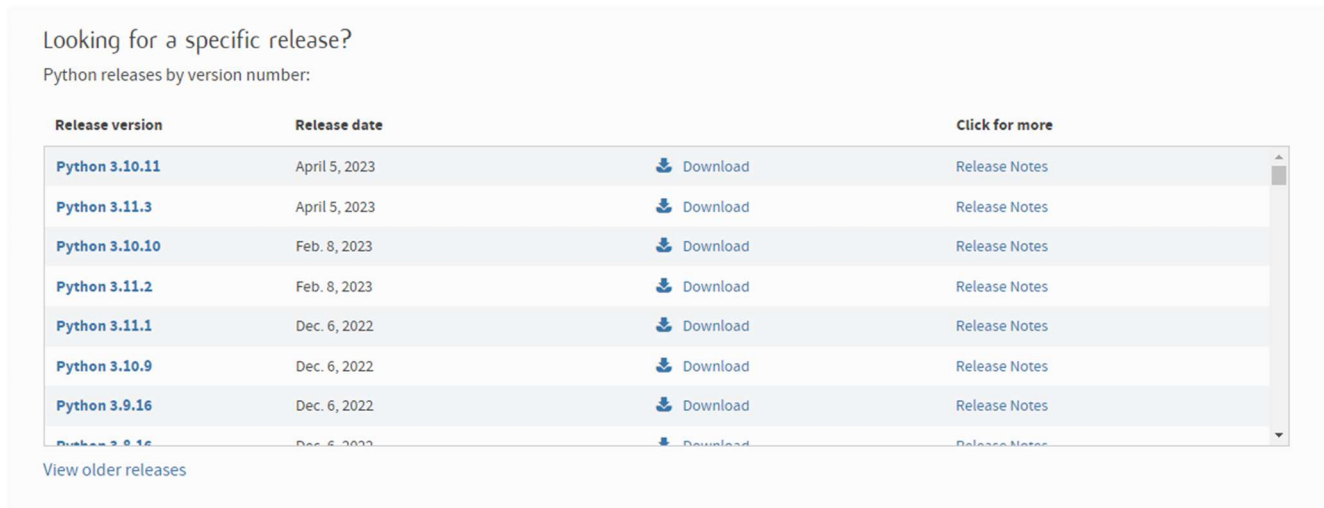


Рис. 1.27 - Версії Python на офіційному сайті

2. Завантажити інсталяційний файл для останньої стабільної версії Python, що відповідає операційній системі, на якій розробляється проект (зазвичай це файл із розширенням .exe).
3. Запустити завантажений інсталяційний файл.
4. Переконаватися, що ввімкнено опцію "Add Python to PATH" (додати Python у змінну середовища PATH). Це дасть змогу використовувати Python із командного рядка без необхідності вказувати повний шлях до виконуваного файлу.
5. Клацнути по кнопці "Install Now" (встановити зараз) і дочекайтеся завершення процесу встановлення.
6. Перевірити успішність встановлення, відкривши командний рядок і ввівши команду "python". Якщо Python успішно встановлено, можна буде побачити інформацію про версію Python та інтерактивну оболонку Python.

```
C:\Users\forcr>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Рис. 1.28 - Демонстрація того, що мова python встановлена

Для комфортної роботи з файлами розширення .ру було прийняте рішення встановити IDE VS Code.

1. Перейти на офіційний сайт Visual Studio Code за адресою <https://code.visualstudio.com/>.
2. На головній сторінці сайту натиснути на кнопку "Download for Windows".
3. Завантажитися інсталяційний файл VS Code у форматі .exe. Щойно завантаження завершиться, знайти цей файл і запустити його.
4. Висвітиться вікно встановлення VS Code. Клацніть на кнопці "Next" (Далі), щоб продовжити.
5. У наступному вікні буде ліцензійна угода. Прочитати її і, якщо умови підходять, необхідно вибрати опцію "I accept the agreement" (Я приймаю умови ліцензійної угоди), потім натиснути кнопку "Next" (Далі).
6. Вибрати шлях для встановлення VS Code або залишити його за замовчуванням. Потім натиснути "Next" (Далі).
7. У наступному вікні вибрати компоненти, необхідно встановити. Зазвичай рекомендується залишити обраними всі компоненти за замовчуванням. Натиснути "Next" (Далі).
8. Потім вибрати папку, в яку будуть додані ярлики VS Code і натиснути "Next" (Далі).
9. Встановлення VS Code на комп'ютер почнеться. Дочекатися завершення процесу.
10. Після завершення встановлення буде вікно з інформацією про завершення. Позначте прапорець "Launch Visual Studio Code" (Запустити Visual Studio Code) і натиснути "Finish" (Готово).

1.5.3 Структура кореневої папки проекту та встановлення бібліотек

Була створена папка, в якій і будуть знаходитися файли проекту. Проект складається з декількох файлів та однієї папки:

					РП 06. 02 001. 00 ДП ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

images	09.03.2023 16:57	Папка с файлами
config.py	17.05.2023 18:42	Python File
db_functions.py	04.06.2023 3:41	Python File
literature_functions.py	04.06.2023 1:38	Python File
main.py	17.05.2023 22:10	Python File
requirements.txt	03.06.2023 22:11	Текстовый докум...
structs.py	03.06.2023 22:11	Python File
test.py	04.06.2023 4:00	Python File

Рис. 1.29 – Структура кореневої папки проекту

- Папка images відповідає за збереження зображень обкладинок літератури для послідууючої відправки користувачу.
- Файл config.py відвідає за збереження токєну боту та словників з відповідями бота на різних мовах.
- Файл db_functions.py відвідає за функції, котрі звертаються до бази даних.
- Файл literature_functions.py відвідає за функції, котрі зв'язані з запитами до джерел з літературою
- Файл main.py відвідає за основний код, з якого запускається бот
- Файл requirements.txt відвідає за збереження списку бібліотек використаних в проекті
- Файл structs.py відвідає за збереження структур для різних жанрів літератури
- Файл test.py потрібен для того, щоб не додаючи код одразу до основних файлів протестувати працездатність наприклад нової функції і вже після того, як буде достовірно відомо, що функція працює і не має багів, тоді вона додається до основних файлів.

Файли передають змінні, функції та класи між собою через import. Ось наприклад import'и, які є в файлі main.py:

```
import telebot
from telebot import types
import requests
```

```

from literature_functions import get_manga_id, get_manga_names, get_image_link,
get_manga_chapters, mangapoisk_last_chapter, is_not_completed
from structs import Manga, Novel
from db_functions import add_to_database_manga, add_to_database_user,
list_of_subscribes, add_manga_to_user
from config import TOKEN
import time
import threading
import pymysql

```

Для встановлення бібліотек, описаних у проекті, було прийняте рішення створити файл requirements.txt. Для встановлення необхідних бібліотек через файл requirements.txt у проекті необхідно виконати такі кроки:

1. Створити файл з іменем requirements.txt у кореневій папці проекту.
2. Відкрити файл requirements.txt за допомогою текстового редактора і додати в нього назви необхідних бібліотек, кожна на окремому рядку.

```

test > requirements.txt
1 telebot
2 requests
3 time
4 threading
5 pymysql
6 requests
7 bs4

```

Рис. 1.30 - Вміст файлу requirements.txt

3. Зберегти файл requirements.txt.
4. Відкрити командний рядок або термінал і перейти у директорію вашого проекту.
5. Переконатися, що встановлено pip, інструмент управління пакетами Python. Його наявність можна перевірити, виконавши команду:

```
pip --version
```

Якщо команда не розпізнається, потрібно встановити pip.

6. Встановити необхідні бібліотеки, виконавши таку команду:

```
pip install -r requirements.txt
```

Ця команда дозволить pip прочитати файл requirements.txt і автоматично встановити всі перераховані в ньому бібліотеки та їхні залежності.

7. Дочекатися завершення процесу встановлення. Pip завантажить і встановить кожен бібліотеку, зазначену у файлі requirements.txt.

1.5.4 Створення класів для відстежуваних витворів

Для короткострокового зберігання даних про мангу та новели, перед їх додаванням до бази даних були розроблені їх класи, а також функції цих класів:

class Manga:

```
last_chapter_ru : list
```

```
last_chapter_en : list
```

```
last_chapter_ua : list
```

```
last_chapter_mangapoisk : list
```

```
manga_name : str
```

```
image_link : str
```

```
manga_id : str
```

```
def set_values(self, last_chapter_ru, last_chapter_en, last_chapter_ua,  
last_chapter_mangapoisk, manga_name, image_link, manga_id):
```

```
self.last_chapter_ru = last_chapter_ru
```

```
self.last_chapter_en = last_chapter_en
```

```
self.last_chapter_ua = last_chapter_ua
```

```
self.last_chapter_mangapoisk = last_chapter_mangapoisk
```

```
self.manga_name = manga_name
```

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

```
self.image_link = image_link
```

```
self.manga_id = manga_id
```

```
class Novel:
```

```
    last_chapter_number : float
```

```
    last_chapter_link : str
```

```
    novel_name : str
```

```
    novel_link : str
```

```
    def set_values(self, last_chapter_number, last_chapter_link, novel_name, novel_link):
```

```
        self.last_chapter_number = last_chapter_number
```

```
        self.last_chapter_link = last_chapter_link
```

```
        self.novel_name = novel_name
```

```
        self.novel_link = novel_link
```

1.5.5 Приклад функцій бота

Починається робота з ботом з команди /start. Після неї бот відправляє таке повідомлення:



Рис. 1.31 - Робота команди /start в моєму боті

Відкривається наступне меню:

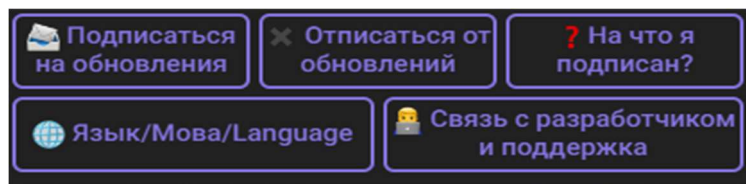


Рис. 1.32 - Меню

За це відповідає цей шматок коду:

```
@bot.message_handler(commands=['start'])
def send_welcome(message):
    lang = 'ru'
    user_id = str(message.chat.id)
    add_to_database_user(user_id)
    menu = main_menu(lang)
    bot.reply_to(message, messages[lang]['greeting'], reply_markup=menu)
```

Який звертається до функцій:

add_to_database_user(), main_menu(). Ось що ці функції з себе представляють:

```
def add_to_database_user(chat_id):

    try:
        connection = pymysql.connect(host = "ЗАСЕКРЕЧЕНО", port = 3306, user =
"root", password="ЗАСЕКРЕЧЕНО", database = "literaturebotdb", cursorclass =
pymysql.cursors.DictCursor)
        with connection.cursor() as cursor:
            select_query = f"SELECT * FROM users WHERE telegram_id = {chat_id}"
            cursor.execute(select_query)
            if cursor.fetchall() != ():
                return None)
            else:
                insert_query = f"INSERT INTO users (telegram_id) VALUES({chat_id})"
                cursor.execute(insert_query)
                connection.commit()

    finally:
        connection.close()
```

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

та

```
def main_menu(lang):  
    menu = types.ReplyKeyboardMarkup(resize_keyboard=True)  
    menu_btn1 = types.KeyboardButton('📧 ' + messages[lang]['subscribe'])  
    menu_btn2 = types.KeyboardButton('✕ ' + messages[lang]['unsubscribe'])  
    menu_btn3 = types.KeyboardButton('? ' + messages[lang]['subscribe_list'])  
    menu_btn4 = types.KeyboardButton('🌐 ' + messages[lang]['language'])  
    menu_btn5 = types.KeyboardButton('👤 ' + messages[lang]['contact'])  
    menu.add(menu_btn1, menu_btn2, menu_btn3, menu_btn4, menu_btn5)  
    return menu
```

Коли користувач вводить команду /start, то бот бере його telegram id та дає його на обробку функції add_to_database_user(). Вона, в свою чергу, спочатку перевіряє, чи є користувач у таблиці `users`. Якщо він там є, то функція не робить нічого, якщо користувача там ще нема, то функція додає id користувача до бази даних. Далі функція send_welcome() відправляє користувачу повідомлення та виводить меню, яке описано в функції main_menu(). У рамках дослідження спробуємо підписатися на той самий комікс, котрий був розглянутий в аналітичному розділу при пошуку аналогів.

Після натискання кнопки “Підписатися на оновлення” та вибору жанру літератури “манга” (на рисунках написи російською мовою, однак у боті передбачена можливість змінити мову) вводиться назва твору

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43



Рис. 1.33 - Пошук творів у боті

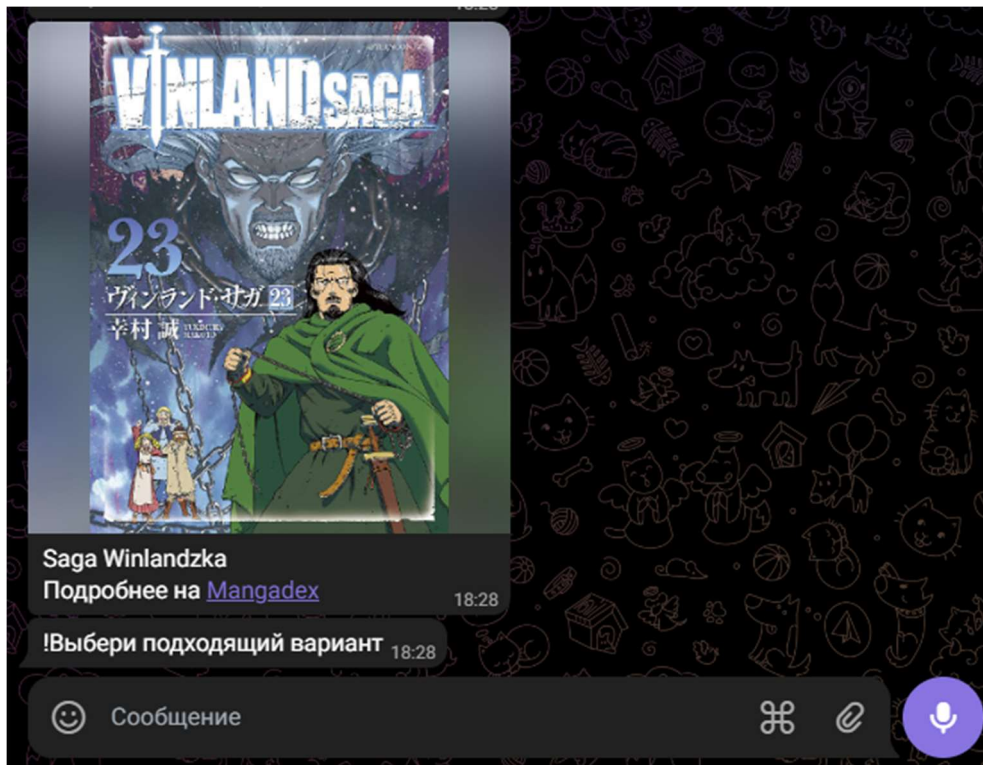


Рис. 1.34 - Вибір варіанту у боті

Бот пропонує одразу декілька варіантів і видає наступне меню:

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

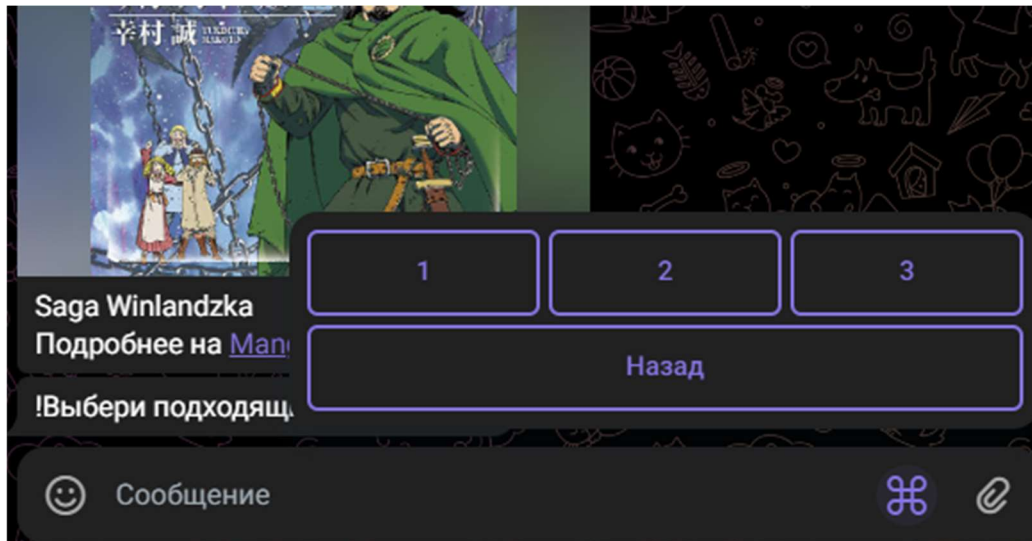


Рис. 1.35 - Меню вибору варіанту у боті

Комікс, який нас цікавить був першим у списку, тому натискаємо кнопку 1:

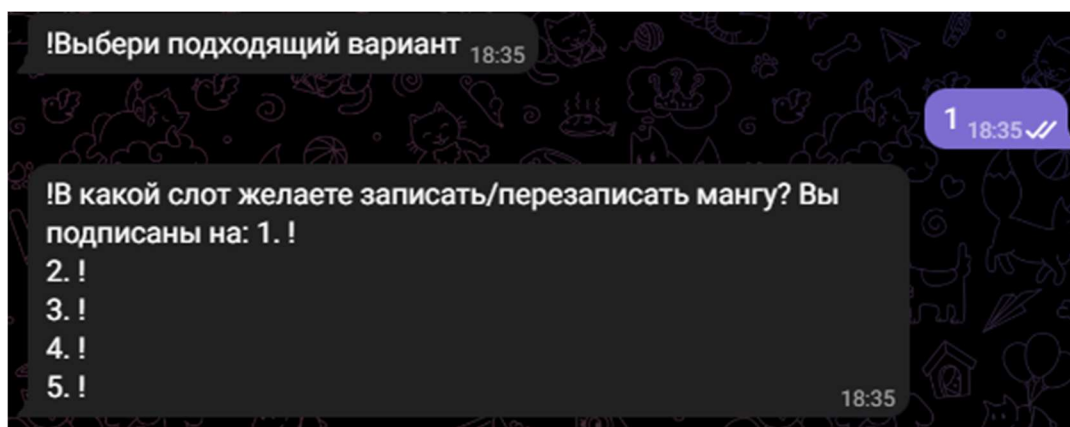


Рис. 1.36 - Вибір слоту у який запишеться витвір

Користувач повинен обрати слот для запису. Всього користувач може відстежувати 5 різних коміксів. Для новел передбачені інші слоти. Користувачу виводиться інше меню:



Рис. 1.37 - Меню вибору слоту.

Був вибраний 1 слот:

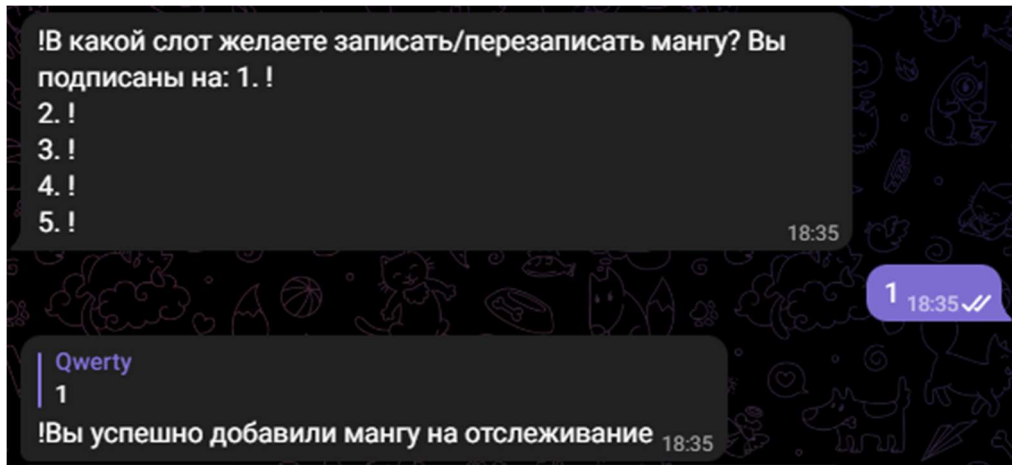


Рис. 1.38 – У рамках дослідження був вибраний 1 слот

Перевірка бази даних з користувачами:

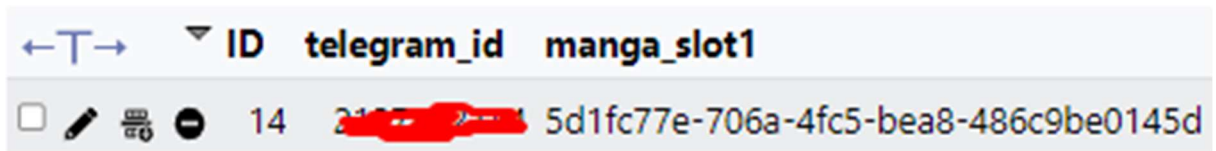


Рис. 1.39 - Запис у таблиці `users`

Користувач додався до бази даних і додав собі у перший слот серед манг ід потрібного коміксу. Для того, щоб переконатися, що в слоті не просто набір символів, а щось осмислене, копіюємо цей ід та підставимо в URL [https://mangadex.org/title/\[ПІДСТАВЛЯЮМО СЮДИ\]](https://mangadex.org/title/[ПІДСТАВЛЯЮМО СЮДИ]):



Рис. 1.40 - Демонстрація того, що ід твору зберігся вірний

					РП 06. 02 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Ця функція бота є обробником повідомлень, який реагує на повідомлення, що починаються з символу '📧'.

Коли користувач надсилає таке повідомлення, викликається функція `subscribe(message)`. Ця функція витягує ідентифікатор чату (`chat_id`) з повідомлення і надсилає повідомлення користувачеві з проханням ввести назву манги. Потім, за допомогою `bot.register_next_step_handler(message, save_title)`, вона реєструє наступний обробник, який буде викликаний після введення користувачем назви манги.

Функція `save_title(message)` отримує введену користувачем назву манги (`manga_title`). Потім вона викликає функцію `get_manga_id(manga_title)`, яка повертає список найкращих результатів (ідентифікаторів манги) для цієї назви манги.

Якщо список `best_results` не порожній, то функція проходить по кожному результату і виконує такі дії:

- Отримує ідентифікатор манги (`manga_id`) зі списку `best_results`.
- Використовуючи функцію `get_manga_names(manga_id)`, отримує назви манги різними мовами.
- Використовуючи `get_image_link(manga_id)`, отримує посилання на зображення манги.
- За допомогою бібліотеки `requests` завантажує зображення манги.
- Надсилає зображення користувачеві з назвою манги і посиланням на `Mangadex`.
- Закриває файл зображення.

Після надсилання зображень та інформації про мангу, функція надсилає користувачеві меню вибору варіанта і реєструє наступний обробник `choise_menu(message)`.

Функція `choise_menu(message)` отримує обраний варіант від користувача. Якщо обрано опцію "назад", то надсилається головне меню. В іншому разі, функція перевіряє, чи можна відстежувати цю мангу (якщо вона завершена, то не можна).

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо мангу можна відстежувати, функція відправляє користувачеві меню вибору слота і реєструє наступний обробник `add_manga(message)`.

Функція `add_manga(message)` отримує обраний слот від користувача. Якщо обрано опцію "назад", то надсилається головне меню. В іншому разі, функція зберігає інформацію про мангу, додає її в базу даних і додає мангу до списку відстежуваних для користувача. Потім надсилається повідомлення про успішне додавання манги і надсилається головне меню.

І так, як невідомо, коли вийде новий розділ цього коміксу, було вирішено зайти до бази даних, та насильно поміняти клітинку `'last_chapter_number'` на 0 і подивитися, чи спрацює відстежування.

Спочатку знаходимо необхідний створець в таблиці `'manga'`:



manga_id	manga_name
af1a-4b5b-a34b-fee2b4cad542	
bd6d0982-0091-4945-ad70-c028ed3c0917	Mushoku Tensei ~Isekai Ittara Honki Dasu~
d1f8fd50-3e24-4397-9b03-a4aa81f3475a	Jishou F-Rank no Onisama ga Game de Hyouka sareru...
30196491-8fc2-4961-8886-a58f898b1b3e	Berserk of Gluttony
5d1fc77e-706a-4fc5-bea8-486c9be0145d	Vinland Saga

Рис. 1.41 – Пошук потрібного твору у таблиці `'manga'`

Дивимось номер останньої глави:



last_chapter_number	last_chapter_link
83	https://mangadex.org/chapter/874631bf-1065-41c6-8d...
36	https://mangapoisk.com/manga/dear-self-styled-f-ra...
54.2	https://mangapoisk.com/manga/berserk-of-gluttony/c...
202	https://mangapoisk.com/manga/vinland-saga/chapter/...

Рис. 1.42 - Номери останніх глав для різних творів. Той, що нас цікавить у рамках цього дослідження є останнім

Далі міняємо цифру 202 в таблиці на 0, через SQL-запрос UPDATE:
UPDATE `mangas` SET `last_chapter_number` = 0 WHERE `manga_id` = '5d1fc77e-706a-4fc5-bea8-486c9be0145d'

І перевіряємо, чи відправить бот через хвилину (спеціально для цього тесту, час засинання циклу у коді був зменшений). І дійсно, він відправив новий розділ:

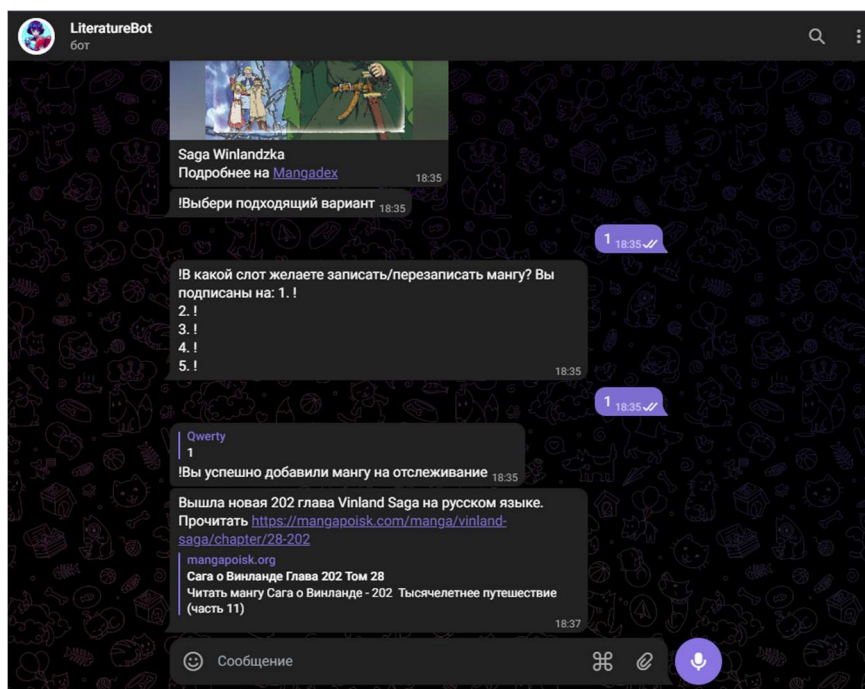


Рис. 1.43 - Бот успішно відправив новий розділ

Після натискання на посилання для прочитання було доведено, що бот функціонує згідно з технічним завданням:

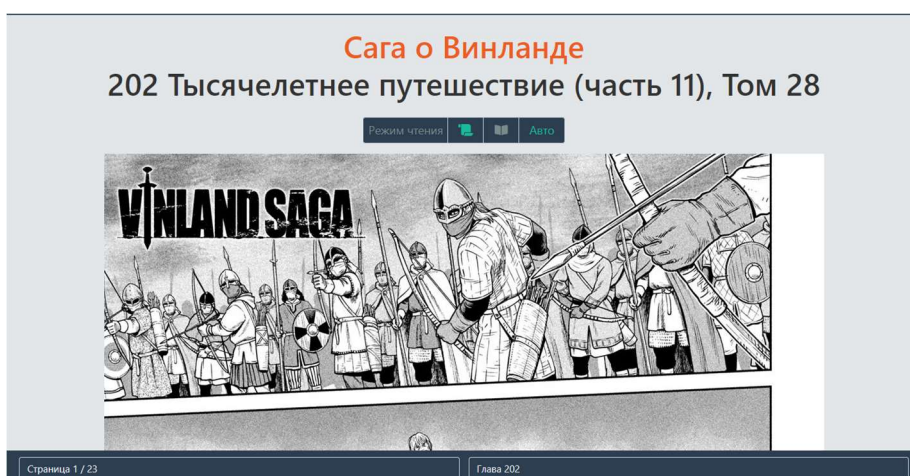


Рис. 1.44 - Демонстрація працездатності посилання відправленого ботом.

					РП 06. 02 001. 00 ДП ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Технічні деталі реалізації бота включають:

- Використання Telegram Bot API: Бот був розроблений з використанням офіційного Telegram Bot API, що дає змогу взаємодіяти з користувачем і обробляти повідомлення.
- Інтеграція із зовнішніми ресурсами: Бот взаємодіє із зовнішніми сервісами, як-от бази даних з інформацією про твори, API для отримання інформації про нові глави та обкладинки, а також ресурси для читання творів.
- Реалізація команд і обробників: Бот обробляє команди від користувачів, як-от підписка на твір і керування підписками. Для кожної команди реалізовано відповідний обробник, який виконує необхідні дії.
- Взаємодія з базою даних: Бот зберігає інформацію про підписки користувачів та інші необхідні дані в базі даних для подальшого використання.

У результаті створення та розгортання цього Telegram-бота користувачі отримали зручний інструмент для відстеження нових розділів сучасної літератури, а автор дипломної роботи продемонстрував навички розроблення, інтеграції із зовнішніми сервісами та управління даними, а також здатність створювати корисні та функціональні додатки.

					<i>РП 06. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

2 ЕКОНОМІЧНА ЧАСТИНА

2.1 Резюме

В даному дипломному проекті розроблений Telegram-бот, призначений для відстеження нових глав сучасної літератури. Мета створення бота - надати користувачам зручний інструмент для отримання інформації про нові випуски та останні оновлення своїх улюблених творів.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних.

Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2. Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога

Каталог аналогів

Таблиця 2.1

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних командах.
1. ПП автоматизованих розрахунків	1300 – 8600
2. ПП введення інформації	1060 – 5750
3. ПП оптимізації розрахунків	1300 – 4200

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

					<i>РП 06. 02 002. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.2.2

Таблиця.2.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7\div 0,8$): $T^a = 229 \times 0,8 = 183,2$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{PP} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.2.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Таблиця 2.2. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
T3 (L_1)	0,15	0,12	0,12
TP (L_2)	0,16	0,15	0,11
PP (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.3. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T^a * L_1 * K_H = 183,2 * 0,12 * 0,9 = 19,78 \quad (\text{люд/годин}) \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T^a * L_2 * K_H = 183,2 * 0,15 * 0,9 = 24,73 \quad (\text{люд/годин}) \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{рп} = T^a * L_3 * K_H * K_T = 183,2 * 0,58 * 0,9 * 0,8 = 76,5 \quad (\text{люд/годин}) \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ} = 3$ (стр), розробка ТП $N_{ТП} = 28$ (стр), розробка робочого проекту $N_{рп} = 17$ (стр), пояснювальна записка відповідно $N_{пз} = 48$ (стр)
Розрахунок зведений у таблицю 4.5

Таблиця 2.5. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1	2	3	4
1.Розробка ТЗ	$T_{РТЗ}=19,78$	$T_{КК}=0,7*N_{ТЗ}=0,7*3=2,1$	$T_{НК}=0,15*N_{ТЗ}=0,15*3=0,45$
2.Розробка ТП	$T_{РТП}=24,73$	$T_{КК}=0,7*N_{ТП}=0,7*28=19,6$	$T_{НК}=0,15*N_{ТП}=0,15*28=4,2$
3.Розробка РП	$T_{РРП}=76,5$	$T_{КК}=0,7*N_{РП}=0,7*17=11,9$	$T_{НК}=0,15*N_{РП}=0,15*17=2,55$
4.Розробка ПЗ	$T_{ПЗ}=1,5**N_{ПЗ}=1,5*47=70,5$	$T_{КК}=0,7*N_{ПЗ}=0,7*48= 33,6$	$T_{НК}=0,15*N_{ПЗ}=0,15*48=7,2$
Усього, в т.ч.:	$\textcircled{C}T=273,1$		
- на розробку	$\textcircled{C}T_p=191,5$		
- контроль керівника		$\textcircled{C}T_{КК}= 67,2$	
- нормоконтроль			$\textcircled{C}T_{НК}=14,4$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 4.6. Відповідно до статті 8 «Закону про Державний бюджет України на 2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року - 6700 гривень; мінімальну погодинну тарифну ставку – 40.46 грн.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	191,5	40.46	7 748.09
2.Контроль керівника	67,2	60.10	4 038.72
3.Нормоконтроль	14,4	60.10	865.44
Усього	-	-	$\textcircled{C}Z_o= 12 643.25$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в

					РП 06. 02 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

таблицю 2.7

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	77	3	231
Транспортні витрати (10%)			23.10	$V_{тр\ з} = 0,1 \times V_{м1} =$
Усього			254.10	$V_M = V_{M1} + V_{тр\ з} =$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.8.

Таблиця 2.8. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	254.10	V_M
2. Основна заробітна плата	12 643.25	Z_o
3. Додаткова заробітна плата	1 896.48	$Z_d = 0,15 \times Z_o =$
4. Відрахування до єдиного фонду соціального внеску	3 198.74	$V_{е.с.в.} = 0,22 \times (Z_o + Z_d)$
5. Накладні витрати	3 792,97	$V_{нак.} = 0,3 \times Z_o$
6. Повна собівартість	21 785.54	$C_{пов} = V_M + Z_o + Z_d + V_{е.с.в.} + V_{нак.} =$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{пов} * P) / 100 \% = (21\ 785.54 * 12,5) / 100 = 2\ 723.19 \quad (2.4)$$

Де P – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{пов} + П = 21\ 785.54 + 2\ 723.19 = 24\ 508.73 \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$ПДВ = 0.2 * Ц_o = 4\ 901,74 \quad (2.6)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 24\ 508.73 + 4\ 901,74 = 29\ 410.47 \quad (2.7)$$

					РП 06. 02 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

3 ОХОРОНА ПРАЦІ

Основним завданням охорони праці є запобігання травмам, професійним захворюванням та іншим шкідливим впливам на здоров'я працівників. Законодавство України, зокрема Конституція України, Закон України "Про охорону праці", основи трудового законодавства та інші законодавчі акти містять вимоги щодо забезпечення безпеки працівників та встановлюють обов'язки для роботодавців з питань охорони праці.

Згідно зі Законом України "Про охорону праці", роботодавець зобов'язаний забезпечити безпечні умови праці для своїх працівників та запровадити необхідні заходи щодо охорони праці. Вимоги до охорони праці повинні бути враховані при розробці проекту будь-якого виробництва, а також при експлуатації обладнання та засобів виробництва.

Основні завдання з охорони праці полягають у забезпеченні безпеки і здоров'я працівників на робочому місці. До них належить:

1. Виявлення можливих небезпек і ризиків на робочому місці і встановлення заходів щодо їх усунення або зменшення.
2. Забезпечення безпечних технологічних процесів праці.
3. Забезпечення безпеки виробничого середовища: організація вентиляції, освітлення, звукоізоляції та інших факторів, що впливають на здоров'я працівників.
4. Забезпечення працівників необхідними засобами індивідуального захисту.
5. Організація навчання та інструктажу працівників з питань охорони праці.

Розробка заходів з охорони праці є необхідною складовою дипломного проекту, оскільки допомагає забезпечити безпеку працівників та виконати вимоги законодавства.

Для програмістів основним положенням щодо їх безпеки є ДСанПіН 3.3.2.-

					<i>РП 06. 02 003. 00 ДП ПЗ</i>	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

007 – 98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями.

Основні вимоги включають такі пункти:

1. Розміщення робочого місця повинно забезпечувати відсутність ближнього розташування екранного пристрою від джерел світла, що можуть викликати блиск.
2. Візуальний дисплейний термінал повинен бути зорієнтований відносно вікон, щоб уникнути падіння прямого сонячного світла на екран, що може викликати блиск.
3. Яскравість екрана має бути налаштована на рівень, що забезпечує комфортні умови роботи, з урахуванням особливостей конкретного виду діяльності.
4. Інтервали роботи з екранним пристроєм повинні бути регулярними і містити перерви для відпочинку очей, що дозволяє зменшити вплив шкідливих факторів.
5. Розміщення робочих місць повинно забезпечувати необхідне просторове розташування обладнання та відповідну організацію робочого місця.
6. Робоче місце повинно бути оснащено стільницею, що дозволяє забезпечити комфортні умови для роботи з екранними пристроями.

3.1 Аналіз небезпечних та шкідливих чинників на робочому місці

Під час експлуатації устаткування можуть мати місце різноманітні фактори, що можуть бути шкідливими та небезпечними для працівників.

Зокрема, можуть бути присутні такі фізичні фактори, як висока температура, надмірний шум, вібрація, іонізуюче та неіонізуюче випромінювання. Також можуть бути присутні хімічні фактори, такі як отруйні речовини, пил, гази та пари,

					РП 06. 02 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

а також біологічні фактори, такі як мікроорганізми, що можуть бути небезпечними для здоров'я працівників.

Психофізіологічні фактори, такі як стрес та психічне перенапруження, також можуть мати негативний вплив на працівників. Для оцінки ступеня впливу кожного фактора на працівника використовуються різноманітні методи та дослідження, такі як аналіз робочого місця, медичні огляди працівників та інші.

3.2 Гігієнічні вимоги до виробничого середовища

Мікроклімат виробничих приміщень повинен відповідати нормативам з температури, вологості, швидкості руху повітря та іншим параметрам, щоб забезпечити комфортні умови для працівників. Освітлення повинно бути достатнім та рівномірним для запобігання зоровому перенапруженню та психоемоційному напруженню.

Шум та вібрації повинні бути зменшені до безпечного рівня для запобігання пошкодженню слуху та здоров'я працівників. Ультразвук та інфразвук повинні бути на безпечному рівні, щоб запобігти шкоді для слуху та здоров'я працівників. Виробничі випромінювання повинні відповідати допустимим рівням, щоб запобігти шкоді для здоров'я працівників.

Електричні системи та обладнання повинні відповідати вимогам безпеки, щоб запобігти ураженню електричним струмом. Усі ці параметри повинні бути забезпечені для забезпечення безпеки та здоров'я працівників.

Для безпечної роботи виробничих приміщень із точки зору мікроклімату, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань та небезпеки ураження електричним струмом, необхідно дотримуватись наступних безпечних рівнів:

1. Освітлення: освітленість на робочій поверхні – не менше 300 лк, на вхідних дверях та проходах – не менше 100 лк.
2. Шум: рівень шуму на робочому місці – не більше 60 дБ.
3. Вібрація: вібрація на робочому місці – не більше 1,5 м/с².

					РП 06. 02 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

4. Ультразвук: рівень ультразвуку – не більше 140 дБ.
5. Інфразвук: рівень інфразвуку – не більше 90 дБ.
6. Виробничі випромінювання: доза випромінювання – не більше 0,1 рентгена за зміну.
7. Небезпека ураження електричним струмом: напруга на дотик – не більше 42 В.

3.3 Вимоги до організації робочого місця працівника

При організації робочого місця , яке передбачає роботу з персональним комп'ютером (ПК) та периферійними пристроями (ПП) (клавіатура, маніпулятор «миша», дискова система, модем, принтер, сканер тощо), слід передбачити:

- ❖ Достатній простір для працівника – площу та простір робочого місця;
- ❖ Площа робочого місця для операторів ЕОМ має бути не менше 6 кв. М. а об'єм – 29 куб м;
- ❖ Ширина робочого місця для операторів ЕОМ має бути не менше 1,5 м;
- ❖ Висота стіни на робочому місці має бути не менше 3,2 м.
- ❖ Розміщення робочого місця по відношенню до світлових прорізів;
- ❖ Вікна мають бути розташовані на рівні, вищому за очі оператора ЕОМ;
- ❖ Необхідно забезпечити захист від прямих променів сонця;
- ❖ Відстані між робочими столами з ПК;
- ❖ Відстань між передніми краями робочих столів із ПК має бути не менше 1,2 м;
- ❖ Відстань між бічними поверхнями моніторів (екранів) має бути не менше 1,5 м;
- ❖ Екран монітора має розташовуватися на відстані 600-700 мм від очей оператора, клавіатура має бути розташована на поверхні столу на відстані 100-300 мм від краю столу.

					<i>РП 06. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

Всі параметри при організації робочого місця відповідають зазначеним вимогам.

3.4 Пожежна безпека

На робочому столі можуть перебувати легкозаймисті матеріали, такі як папір, ручки, канцелярські клеї та інше. У кімнаті також є шафа і диван. У шафі знаходиться легкозаймистий одяг. На дивані можуть бути матеріали, здатні швидко спалахнути в разі прямого потрапляння на них джерела тепла.

Для визначення категорії приміщення за вибухопожежною небезпекою та вибору засобів пожежогасіння і запобігання, необхідно проводити комплексне технічне обстеження приміщення та його вмісту згідно з відповідними стандартами і нормативами.

Загалом, для забезпечення безпеки в приміщенні необхідно дотримуватися таких правил:

1. Не зберігати легкозаймисті матеріали поруч із джерелами тепла;
2. Забезпечити правильне зберігання хімічних речовин, дотримуватися вимог щодо їх маркування та використання;
3. Регулярно проводити перевірки та технічні обслуговування всіх електричних пристроїв, а також обладнання для пожежогасіння та попередження.

Для більшої безпеки будівля та приміщення повинні бути оснащені первинними засобами пожежогасіння.

Під час роботи працівники можуть бути піддані різноманітним фізичним, хімічним та біологічним факторам, які можуть мати шкідливий вплив на їх здоров'я. Оцінка ступеня впливу цих факторів та заходи захисту від них мають велике значення для забезпечення безпеки працівників на робочому місці. Також не можна випускати з виду і пожежну небезпеку. Для захисту від цих факторів використовуються різноманітні технології, устаткування та особисті засоби захисту, а також існують нормативні документи, які встановлюють допустимі та граничні значення дії цих

					РП 06. 02 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

факторів. Виконання заходів забезпечення безпеки на робочому місці є важливим елементом ефективного управління підприємством та покращення умов праці працівників. Я думаю, що охорона праці в сучасному виробництві дуже важлива, вона забезпечує безпеку та здоров'я працівників, зменшує кількість нещасних випадків та покращує умови праці. Розробка заходів з охорони праці є необхідною складовою дипломного проекту і законодавство України містить вимоги щодо забезпечення безпеки працівників.

					<i>РП 06. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ВИСНОВКИ

У рамках дипломної роботи було розроблено Telegram-бот, призначений для відстеження нових глав сучасної літератури. Метою створення бота було надати користувачам зручний інструмент для отримання інформації про нові випуски та останні оновлення своїх улюблених творів.

Основні функціональні можливості бота включають:

- Підписку на мангу та ранобе: Користувачі можуть підписуватися на твори, які їх цікавлять, вказуючи їхні назви.
- Отримання сповіщень про нові розділи: У разі появи нового розділу бот автоматично повідомляє передплатників, надаючи інформацію про назву твору та доступні розділи.
- Відображення інформації про твір: Бот надає детальну інформацію про кожен твір, включно з назвою, автором, жанром, описом і обкладинкою. Хоча деякі з цих даних доступні лише за посиланням.
- Надання посилань на ресурси: Для кожного твору бот надає посилання на ресурси, де користувачі можуть прочитати нові глави або отримати додаткову інформацію.
- Керування підписками: Користувачі можуть керувати своїми підписками, додавати нові твори, видаляти або змінювати налаштування наявних підписок

У економічній частині дипломного проекту також було визначено ціну програмного продукту, яка склала 29 410.47 грн.

Технічні деталі реалізації бота включають:

- Використання Telegram Bot API: Бот був розроблений з використанням офіційного Telegram Bot API, що дає змогу взаємодіяти з користувачем і обробляти повідомлення.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

- Інтеграція із зовнішніми ресурсами: Бот взаємодіє із зовнішніми сервісами, як-от бази даних з інформацією про твори, API для отримання інформації про нові глави та обкладинки, а також ресурси для читання творів.
- Реалізація команд і обробників: Бот обробляє команди від користувачів, як-от підписка на твір і керування підписками. Для кожної команди реалізовано відповідний обробник, який виконує необхідні дії.
- Взаємодія з базою даних: Бот зберігає інформацію про підписки користувачів та інші необхідні дані в базі даних для подальшого використання.

Також були розглянуті питання щодо охорони праці, де виявилось, що під час роботи працівники можуть бути піддані різноманітним фізичним, хімічним та біологічним факторам, які можуть мати шкідливий вплив на їх здоров'я. Оцінка ступеня впливу цих факторів та заходи захисту від них мають велике значення для забезпечення безпеки працівників на робочому місці. Також не можна випускати з виду і пожежну небезпеку. Для захисту від цих факторів використовуються різноманітні технології, устаткування та особисті засоби захисту, а також існують нормативні документи, які встановлюють допустимі та граничні значення дії цих факторів. Виконання заходів забезпечення безпеки на робочому місці є важливим елементом ефективного управління підприємством та покращення умов праці працівників. Розробка заходів з охорони праці є необхідною складовою дипломного проекту і законодавство України містить вимоги щодо забезпечення безпеки працівників.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Простой Python. Современный стиль программирования. 2-е изд. / Любанович Билл. – "Издательский дом "Питер"", 2020 г. – 592 с.
2. Праворська Н.І. Інформатика та комп'ютерна техніка: Навчально-методичний посібник для студентів вищих навчальних закладів. – Хмельницький, 2002. – 312с
3. Скрапинг веб-сайтов с помощью Python. / Райан Митчелл. – ЛитРес, 2022 г. – 336 с.
4. Копайгородська Т.Г.Методичні вказівки до виконання економічного розділу. ВСП ОТФК ОНТУ, 2023
5. Документація по створенню Telegram-ботів [Електронний ресурс]. URL: <https://core.telegram.org/bots>
6. Грофф Д. SQL: полное руководство / Д. Грофф, П. Вайнберг. – К.: BHV, 2005. – 608 с.
7. Python is a programming language, [Електронний ресурс]. URL: <https://www.python.org>
8. Основы Python. Научитесь думать как программист. / Аллен Б. Дауни. – "Манн, Иванов и Фербер", 2021 г. – 304 с.
9. MySQL. Полн. и исчерпывающее руководство по применению и администрированию баз данных MySQL 4, а также программированию приложений. / Поль Дюбуа, Н.В Воронин – «Вильямс», 2004 г. – 1056 с.
10. Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein – «McGraw-Hill», 2009 г. – 1292 с.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТОК А. Робота нескінченного циклу для відстежування оновлень сучасної літератури

```
while True:
```

```
    connection = pymysql.connect(host="***.*.*.*", port=****, user="root",  
password="*****", database="literaturebotdb",  
cursorclass=pymysql.cursors.DictCursor)
```

```
    with connection.cursor() as cursor:
```

```
        select_query = "SELECT * FROM `mangas`"
```

```
        cursor.execute(select_query)
```

```
        row = cursor.fetchall()
```

```
        connection.commit()
```

```
        for i in row:
```

```
            if float(get_manga_chapters(i["manga_id"])[1][1]) >=  
float(mangapoisk_last_chapter(i["manga_name"])[1]):
```

```
                current_manga = get_manga_chapters(i["manga_id"])
```

```
                if float(current_manga[1][1]) > i["last_chapter_number"]:
```

```
                    update_query = f"UPDATE `mangas` SET `last_chapter_number` =  
{current_manga[1][1]}, `last_chapter_link` =  
'https://mangadex.org/chapter/{current_manga[1][0]}',  
`last_ru_chapter_mangadex_number` = {current_manga[1][1]},  
`last_ru_chapter_mangadex_link` =  
'https://mangadex.org/chapter/{current_manga[1][0]}' WHERE `manga_id` =  
'{i['manga_id']}'"
```

```
                    cursor.execute(update_query)
```

```
                    connection.commit()
```

```
                    select_query = "SELECT * FROM `users`"
```

```
                    cursor.execute(select_query)
```

```
                    row_2 = cursor.fetchall()
```

```
                    connection.commit()
```

```

        for o in row_2:

            if i["manga_id"] == o["manga_slot1"] or i["manga_id"] ==
o["manga_slot2"] or i["manga_id"] == o["manga_slot3"] or i["manga_id"] ==
o["manga_slot4"] or i["manga_id"] == o["manga_slot5"]:

                bot.send_message(o["telegram_id"], f"Вышла новая
{current_manga[1][1]} глава {i['manga_name']} на русском языке. Называется
{current_manga[1][2]}. Прочитать
https://mangadex.org/chapter/{current_manga[1][0]}")

                if float(get_manga_chapters(i["manga_id"])[1][1]) <
float(mangapoisk_last_chapter(i["manga_name"])[1]):

                    current_manga = mangapoisk_last_chapter(i["manga_name"])

                    if float(current_manga[1]) > i["last_chapter_number"]:

                        update_query = f"UPDATE `mangas` SET `last_chapter_number` =
{current_manga[1]}, `last_chapter_link` = '{current_manga[0]}',
`last_chapter_mangapoisk_number` = {current_manga[1]},
`last_chapter_mangapoisk_link` = '{current_manga[0]}' WHERE `manga_id` =
'{i['manga_id']}'"

                        cursor.execute(update_query)

                        connection.commit()

                        select_query = "SELECT * FROM `users`"

                        cursor.execute(select_query)

                        row_2 = cursor.fetchall()

                        connection.commit()

                        for o in row_2:

                            if i["manga_id"] == o["manga_slot1"] or i["manga_id"] ==
o["manga_slot2"] or i["manga_id"] == o["manga_slot3"] or i["manga_id"] ==
o["manga_slot4"] or i["manga_id"] == o["manga_slot5"]:

                                bot.send_message(o["telegram_id"], f"Вышла новая
{current_manga[1]} глава {i['manga_name']} на русском языке. Прочитать
{current_manga[0]}")

                                with connection.cursor() as cursor:

                                    select_query = "SELECT * FROM `novels`"

```

```
cursor.execute(select_query)
row = cursor.fetchall()
connection.commit()
for i in row:
    current_chapter = novela_id_data(i["novel_id"])
    if float(current_chapter[2]) > i["last_chapter_number"]:
        update_query = f"UPDATE `novels` SET `last_chapter_number` =
{current_chapter[2]}, `last_chapter_link` = 'ranobe.me{current_chapter[1]}'
WHERE `novel_id` = '{i['novel_id']}'"
        cursor.execute(update_query)
        connection.commit()
        select_query = "SELECT * FROM `users`"
        cursor.execute(select_query)
        row_2 = cursor.fetchall()
        connection.commit()
        for o in row_2:
            if i["novel_id"] == o["novel_slot"]:
                bot.send_message(o["telegram_id"], f"Вышла новая
{current_chapter[0]} глава {i['novel_name']} на русском языке. Прочитать
{current_chapter[1]}")

time.sleep(900)
```

ДОДАТОК Б. Робота функції для додавання в базу даних нових творів та прив'язки творів до користувачів

```
@bot.message_handler(func=lambda message: message.text.startswith('✉'))
def start_subscribe(message):
    menu = genre_menu(lang)
    bot.send_message(message.chat.id, "!Что вы хотите найти?",
reply_markup=menu)
    bot.register_next_step_handler(message, chocie)

def chocie(message):
    if message.text == "Манга":
        chat_id = message.chat.id
        bot.send_message(chat_id, '!Введите название манги:')
        bot.register_next_step_handler(message, save_title_manga)
    elif message.text == "Новелла":
        chat_id = message.chat.id
        bot.send_message(chat_id, '!Введите название новеллы:')
        bot.register_next_step_handler(message, subscribe_novela)

def subscribe_novela(message):
    global novel_data
    novel_data = novela_all_data(message.text)
    print(message.text)
    print(novel_data)
    print(novela_all_data(message.text))
    if novel_data == None:
        menu = main_menu(lang)
```

```
    bot.reply_to(message, "!Ничего не найдено", reply_markup=menu)
else:
    if novel_data[6] == True:
        menu = main_menu(lang)
        bot.reply_to(message, "!Новелла, которую вы хотите отслеживать
завершена", reply_markup=menu)
    else:
        menu = subscribe_novel_menu(lang)
        chat_id = message.chat.id
        bot.send_message(chat_id, f'{novel_data[4]}\nНазвание:
{novel_data[2]}\nОписание: {novel_data[5]}')
        bot.send_message(chat_id, '!Желаете подписаться на данную новеллу?',
reply_markup=menu)
        bot.register_next_step_handler(message, choice_novela)
```

```
def choice_novela(message):
```

```
    global novel_data
    answer = message.text
    if answer == messages[lang]['back']:
        menu = main_menu(lang)
        bot.reply_to(message, "!Без проблем", reply_markup=menu)
    elif answer == "Подписаться":
        menu = main_menu(lang)
        novel.set_values(novel_data[0], novel_data[1], novel_data[2], novel_data[3])
        add_to_database_novel(novel.last_chapter_number, novel.last_chapter_link,
novel.novel_name, novel.novel_link)
        add_novel_to_user(novel.novel_link, message.chat.id)
        bot.reply_to(message, "!Успешно добавлено", reply_markup=menu)
```

```

def save_title_manga(message):
    manga_title = message.text
    global best_results
    best_results = get_manga_id(manga_title)[:3]
    if best_results != []:
        for i in range(len(best_results)):
            manga_id = get_manga_id(manga_title)[i]
            if get_manga_names(manga_id)[1] != None:
                name = get_manga_names(manga_id)[1]
            elif get_manga_names(manga_id)[2] != None:
                name = get_manga_names(manga_id)[2]
            else:
                name = get_manga_names(manga_id)[0]
            p = requests.get(get_image_link(manga_id))
            out = open(f'D:\Projects\test\images\img{i}.jpg', "wb")
            out.write(p.content)
            out = open(f'D:\Projects\test\images\img{i}.jpg', "rb")
            bot.send_photo(message.chat.id, out, f'{name}\nПодробнее на <a
href="https://mangadex.org/title/{manga_id}">Mangadex</a>',
parse_mode='HTML')
            out.close()
            menu = choose_variant_menu(lang)
            bot.send_message(message.chat.id, "!Выбери подходящий вариант",
reply_markup=menu)
            bot.register_next_step_handler(message, choose_menu_manga)
        else:
            menu = main_menu(lang)
            bot.send_message(message.chat.id, "!Ничего не найдено",
reply_markup=menu)

```

```

def chose_menu_manga(message):
    choice_variant = message.text
    if choice_variant == messages[lang]['back']:
        menu = main_menu(lang)
        bot.reply_to(message, "!Без проблем", reply_markup=menu)
    elif is_not_completed(best_results[int(choice_variant) - 1]):
        global choice_manga
        choice_manga = best_results[int(choice_variant) - 1]
        menu = choose_slot_menu(lang)
        slots = list_of_subscribes(message.chat.id)[0]
        for i in slots:
            if slots[i] == None:
                slots[i] = '!Пусто'
            else:
                slots[i] = get_manga_names(slots[i])

        bot.send_message(message.chat.id, f"!В какой слот желаете
записать/перезаписать мангу? Вы подписаны на: 1.
{slots['manga_slot1'][0]}\n2. {slots['manga_slot2'][0]}\n3.
{slots['manga_slot3'][0]}\n4. {slots['manga_slot4'][0]}\n5.
{slots['manga_slot5'][0]}", reply_markup=menu)

        bot.register_next_step_handler(message, add_manga)
    else:
        menu = main_menu(lang)

        bot.reply_to(message, "!Данная манга завершена, вы не можете ее
отслеживать", reply_markup=menu)

def add_manga(message):
    choice_variant = message.text

```

```
if choice_variant == messages[lang]['back']:
    menu = main_menu(lang)
    bot.reply_to(message, "!Без проблем", reply_markup=menu)
elif int(choice_variant) > 0 and int(choice_variant) < 6:
    languages = get_manga_chapters(choice_manga)
    if get_manga_names(choice_manga)[1] == None:
        link_for_mangapoisk =
mangapoisk_last_chapter(get_manga_names(choice_manga)[0])
    else:
        link_for_mangapoisk =
mangapoisk_last_chapter(get_manga_names(choice_manga)[1])
        manga.set_values(last_chapter_ru = languages[1], last_chapter_en =
languages[0], last_chapter_ua = languages[2], last_chapter_mangapoisk =
link_for_mangapoisk, manga_name = get_manga_names(choice_manga)[0],
image_link = get_image_link(choice_manga), manga_id = choice_manga)
        add_to_database_manga(manga.manga_id, manga.manga_name,
manga.last_chapter_ru[1], manga.last_chapter_ru[0], manga.last_chapter_ua[1],
manga.last_chapter_ua[0], manga.last_chapter_en[1], manga.last_chapter_en[0],
manga.last_chapter_mangapoisk[1], manga.last_chapter_mangapoisk[0])
        add_manga_to_user(choice_manga, message.chat.id, choice_variant)
        menu = main_menu(lang)
        bot.reply_to(message, "!Вы успешно добавили мангу на отслеживание",
reply_markup=menu)
```


ДОДАТОК В. Презентація

Слайд 1

РОЗРОБКА ТЕЛЕГРАМ-БОТУ ДЛЯ ВІДСТЕЖУВАННЯ ОНОВЛЕНЬ СУЧАСНОЇ ЛІТЕРАТУРИ

- Виконав Борисков Д.Є.
гр. 4РП-06

ВСП «Одеський Технічний Фаховий Коледж ОНТУ»

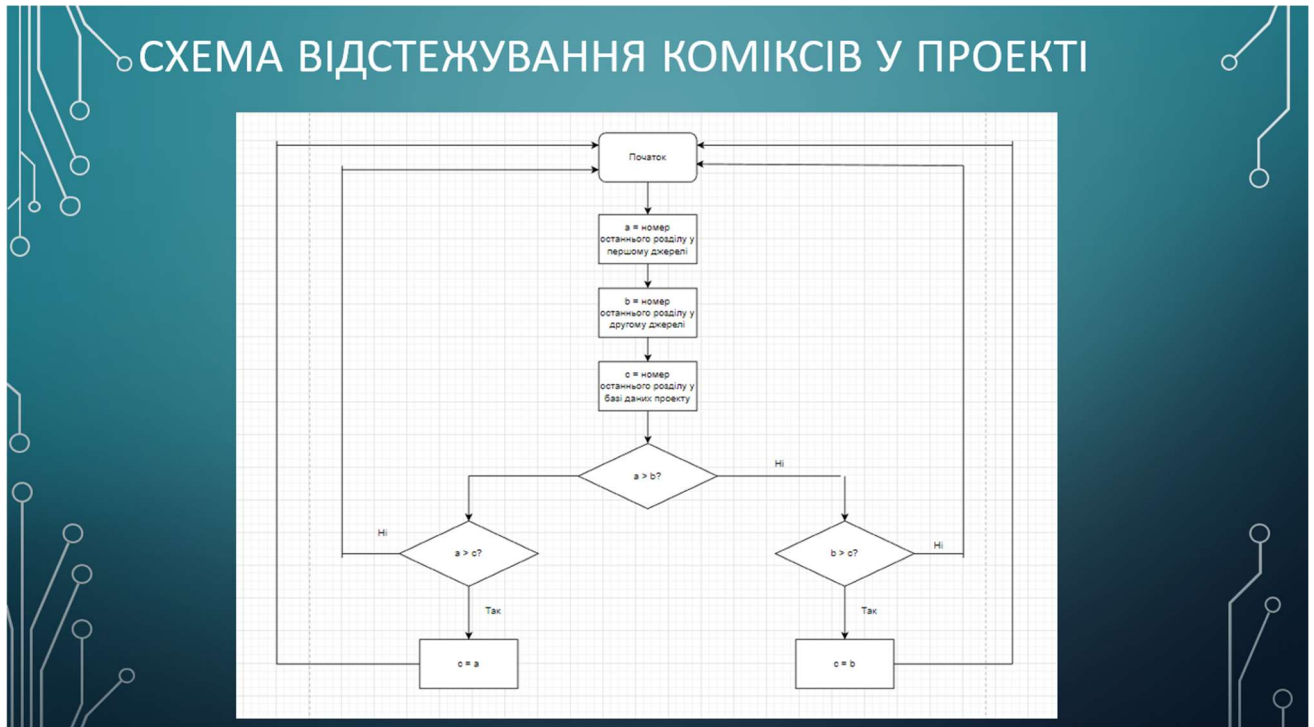


Слайд 2

СТРУКТУРА ПРОЕКТУ

images	09.03.2023 16:57	Папка с файлами
config.py	17.05.2023 18:42	Python File
db_functions.py	04.06.2023 3:41	Python File
literature_functions.py	04.06.2023 1:38	Python File
main.py	17.05.2023 22:10	Python File
requirements.txt	03.06.2023 22:11	Текстовый докум...
structs.py	03.06.2023 22:11	Python File
test.py	04.06.2023 4:00	Python File

Слайд 5



Слайд 6



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Боровецький Юрій Дмитрович

(прізвище, ім'я та по батькові)

Спеціальність: 121 - Інженерія програмного забезпечення

Освітня програма: Розробка програмного забезпечення

Тема дипломного проекту: Розробка клієнтської частини для веб-сайту ВСП «ОТФК
ОНТУ»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки)

Пояснювальна записка розроблена відповідно до нормативів ДСТУ. Обсяг та суть роботи гармонійно відповідають меті, яка була визначена для дипломного проектування. Розділ, що містить розрахунки та пояснення в проекті, відзначається логічною структурою та відмінною якістю реалізації.

б) самостійність роботи над проектом:

Боровецький Ю.Д. з фокусом і цілеспрямованістю зосередився на роботі з науковими матеріалами, провів ґрунтовний аналіз методів та стратегій розробки веб-орієнтованих систем. Він демонстрував грамотне застосування сучасних інформаційних технологій у своїх дослідженнях.

в) теоретична підготовка випускника (випускниці):

Студент Боровецький Ю.Д. має солідну теоретичну та практичну базу на високому рівні. Він проявляє високу концентрацію, пильність та відповідальність у своїй роботі.

г) вміння розв'язувати виробничі та конструкторські питання _____

Студент Боровецький Ю.Д. за період роботи показав Високий рівень теоретичної та
практичної підготовки

Оцінка розрахункової частини «Відмінно»

Оцінка графічної частини «Відмінно»

Загальна оцінка «Відмінно»

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Шувалова Ірина Олегівна

Місце роботи і посада керівника дипломного проекту Викладач

Підпис _____

« 09 » 06 2023 р.

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Боровецький Юрій Дмитрович

(прізвище, ім'я та по батькові)

Спеціальність: 121 - Інженерія програмного забезпечення

Освітня програма: Розробка програмного забезпечення

Тема дипломного проекту: Розробка клієнтської частини для веб-сайту ВСП «ОТФК
ОНТУ»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки)

Пояснювальна записка розроблена відповідно до нормативів ДСТУ. Обсяг та суть роботи гармонійно відповідають меті, яка була визначена для дипломного проектування. Розділ, що містить розрахунки та пояснення в проекті, відзначається логічною структурою та відмінною якістю реалізації.

б) самостійність роботи над проектом:

Боровецький Ю.Д. з фокусом і цілеспрямованістю зосередився на роботі з науковими матеріалами, провів ґрунтовний аналіз методів та стратегій розробки веб-орієнтованих систем. Він демонстрував грамотне застосування сучасних інформаційних технологій у своїх дослідженнях.

в) теоретична підготовка випускника (випускниці):

Студент Боровецький Ю.Д. має солідну теоретичну та практичну базу на високому рівні. Він проявляє високу концентрацію, пильність та відповідальність у своїй роботі.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Боровецький Юрій Дмитрович,
здобувач освіти гр. 4РП-06, та

Шувалова Ірина Олегівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

**«Розробка клієнтської частини для веб-сайту ВСП «ОТФК ОНТУ»»
(автор роботи – Боровецький Ю.Д., керівник роботи – Шувалова І.О.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець

Б.Ю.

/ Боровецький Ю.Д. /

Керівник

Шувалова І.О.

/ Шувалова І.О. /

« 09 » 06 20 23 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015551584

Дата перевірки:
11.06.2023 22:38:34 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
11.06.2023 22:39:32 EEST

ID користувача:
100011688

Назва документа: 4РП-06 Боровецький Ю.Д

Кількість сторінок: 57 Кількість слів: 7742 Кількість символів: 59359 Розмір файлу: 11.72 МВ ID файлу: 1015203857

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.95%

Схожість

Найбільша схожість: 0.63% з Інтернет-джерелом (https://zntu.edu.ua/sites/default/files/konf/konspekt_lekciy_dlya_sam..)

3.95% Джерела з Інтернету

529

Сторінка 59

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

9

Підозріле форматування

9
сторінок

г) перелік позитивних якостей дипломного проекту (роботи) _____

Серед позитивних аспектів проекту можна виділити використання передових технологій, увагу до деталей, високу продуктивність та зручність інтерфейсу.

д) основні недоліки дипломного проекту (роботи) _____

Щодо недоліків, бажано було б бачити більш розгорнуту частину, що присвячена тестуванню і забезпеченню якості.

Оцінка розрахункової частини _____ добре

Оцінка графічної частини _____ добре

Загальна оцінка _____ добре

Прізвище, ім'я, по батькові рецензента _____

Васіліу Євген Вікторович

Місце роботи і посада рецензента _____

Державний університет інтелектуальних технологій

і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ, декан факультету інформаційних технологій та кібербезпеки

Підпис: _____

« 16 » 06 2023 р.

