КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Випускна кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки на тему:

Розробка Телеграм-бота для гри в рулетку

Виконав студент 4 курсу
Олішевський Кирило Віталійович

_____
(підпис)

Науковий керівник:
доцент
Ставровський Андрій Борисович

_____
(підпис)

Засвідчую, що в цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри теоретичної кібернетики
« » червня 2022 р., протокол №
Завідувач кафедри
Ю. В. Крак _____
(підпис)

Київ-2022

ЗМІСТ

РЕФЕРАТ

Обсяг роботи складає 32 сторінок, використано 8 джерело посилань.

Розробка Телеграм-бота для гри в рулетку, робота з базою.

Об’єктом роботи є процес розробки телеграмм бота онлайн-рулетка.

Мета роботи полягає в:

- Створення архітектури та структури проекту
- Встановленні необхідних модулів та компонентів
- Робота з базою
- Написання логіки гри
- Реалізація взаємодії телеграмма и бази даних.
- Реалізація платіжної системи

Інструменти, що використовуються:

- Мова Python
- MYSQL БД
- Telegram API
- QIWI API

Результати роботи: обдумана архітектура телеграмм-бота. Створена база даних та підключена до бота. Реалізовані всі операції з базою для гри. Реалізовані всі алгоритми ставок. Також була зроблена платіжна система для обробки балансу гравців. Отримані результати повністю відповідають темі курсової роботи.

ВСТУП

У Telegram працює платформа чатботів. Боти можуть виконувати різноманітні завдання, такі як пошук в інтернеті чи держреєстрах, покупки, платежі, розваги, модерація груп тощо. У спілкуванні беруть участь користувач Telegram та комп'ютерна програма від стороннього розробника.

Оскільки сучасний світ не стоїть на місці, технології стрімко розвиваються та кожного дня з’являються нові тренди та ідеї, багато з них мають можливість вплинути на IT галузь, яка в свою чергу впливає на інші галузі, що в подальшому впливатиме на наше повсякденне життя.

Якщо до недавнього часу популярними були додатки або комп’ютерні програми, то на даний момент лідерство займають чат-боти, які мають великі перспективи в різних сферах нашого життя.

Для початку потрібно визначити, що представляє собою чат-бот. Чатбот – це спеціалізований додаток, що дозволяє користувачам взаємодіяти зі сторонніми сервісами, якщо існує така необхідність і все це виконано через всім відомий інтерфейс чату.

Чат-бот – це деякий помічник, який спілкується з користувачами через повідомлення і має множину певних функцій. Тобто, можна отримати певну інформацію, написавши чат-боту спеціальну команду, яку в свою чергу останній інтерпретує певним чином.

Так можна швидко переводити, коментувати, знаходити, тестувати, шукати, навчати, транслювати, вбудовуватися в інші сервіси і платформи, взаємодіяти з датчиками і речами, підключеними до інтернету.

Користувач може взаємодіяти з ботом за допомогою елементів інтерфейсу месенджера: надсилання повідомлень, натискання на команди та кнопки, використання інлайн-режиму. Telegram надає три способи взаємодії користувача з ботом: приватний чат (класичний спосіб), група й так званий інлайн-режим:

- Найпоширеніший спосіб — приватний чат. Здебільшого бот не може ініціювати діалог із користувачем.
- Деякі боти можуть бути учасниками груп. Наприклад, у групах бот може підтримувати розмову, модерувати повідомлення або бути ведучим гри.
- Інлайн-режим нагадує інтерфейс пошукової системи. Користувач уводить у поле для введення повідомлень запит, що починається з короткого імені бота. Далі користувач може вибрати й надіслати один з результатів.

Бот взаємодіє з користувачами за допомогою Bot API[1] Керування ботами на платформі доступне в боті BotFather[2]. Зокрема, там можна створити бота та переглянути чи змінити присвоєний йому токен API[3].

Метою і завданням роботи є розробка телеграмм бота за допомогою TELEGRAM BOT API та взаємодія з базою і платіжною системою. **Актуальність роботи та підстави для її виконання.**

Оптимізація та зручність роботи. В даний час більшість людей спілкуються в месенджерах, тому це гарна платформа для розробки. Telegram BOT є гарний аналог будь-якого сайту, тому що можно реалізувати аби-який функціонал , а це є і дешевше і зручніше для користувачів. Тому актуальність даної задачі дуже велика.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Створення Telegram-bot та отримання API Token. Бот взаємодіє з користувачами за допомогою Bot API[1] Керування ботами на платформі доступне в боті BotFather[2]. Зокрема, там можна створити бота та переглянути чи змінити присвоєний йому токен API[3].

Створення бази даних MYSQL. Піднімаємо сервер на localhost та створюємо базу даних та таблиці для подальшої роботи .

Створення платіжної системи. Використовуючи QIWI API та електронний гаманець буде реалізація платіжної системи.

Написання коду на python. Використовуємо бібліотеку telebot для взаємодії с Telegram API. Підключення бази даних, реалізація логіки гри та інтерфейсу для користувача, створення платіжної системи для обробки платежів та балансу гравця.

Функціонал бота. Є гра в рулетку , максимально реалізовано схожість з фізичним аналогом цієї гри.

Реалізован баланс (депозит) який можно ставити та грати. Баланс можно поповнювати різними способами, будь-які платіжні системи, на даний момент підключена платіжна система QIWI.

Баланс кожного гравця зберігається в базі даних. Операції с балансом виконуються через інтерфейс боту користувачем , та обробляється на сервері.

Доступні функції : Поповнення балансу через платіжні системи.

Перевірка балансу за допомогою запиту до бази даних.

Вивід балансу на свій будь який рахунок банку, або електронного гаманця.

Кнопка інформації про короткий опис боту.

Реферальна система реалізовані такі функції:

Дізнатися свій реферальний код для закликання людей , та отримання бонусів.

Бонус от реферальной системи полягає в отриманні 40% від програшу вашего реферала. Тобто якщо ваш реферал програє X- сумму , то ви отримаєте 0.4X .

Кнопка Оновити реферал, дає змогу змінити людину яка вас запросила і яка буде отримувати бонуси.

Також реалізована статистика ваших рефералів, їх кількість та ваша процентна ставка.

Найцікавіший функціонал цього бота являє собою ставки.

Можно ставити будь які суми, з умовою достатнього балансу.

Всього є 6 типу ставок , на які є можливість зіграти в рулетку. (В моєму випадку)

Ставка на Червоне.

Ставка на Чорне.

Ставка на будь-яке число.

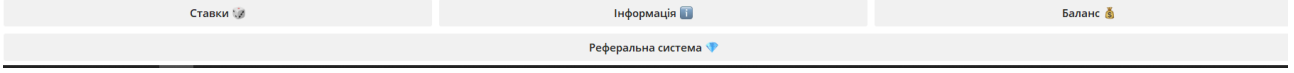
Ставка на парне число.

Ставка на непарне число.

Ставка на будь-яку дюжину.

Інтерфейс: Інтерфейс складається з меню та відео самої гри.

Пункти меню:



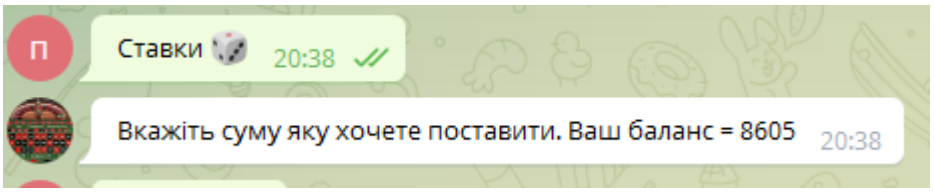
Ставки 🎲

Інформація ℹ️

Баланс 💰

Реферальна система 💎

Інтерфейс прийняття суми ставки:



Варіанти ставок:

Червоне 🔴

Чорне ⚫

Число 🔢

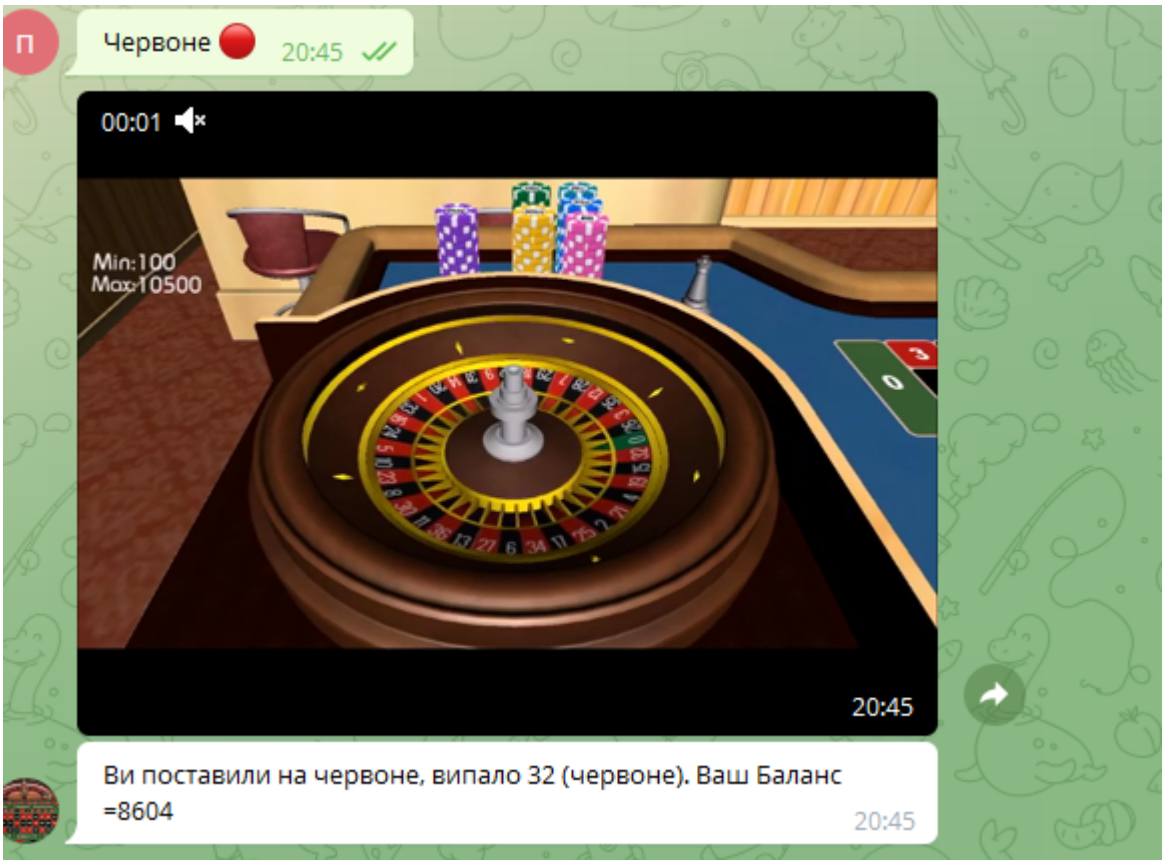
Парне

Непарне

Дюжини

Відміна Ставки

Після вибраної ставки, починається гра. Та показує відео ролик с числом яке випало в рулетці.



Інструменти для розробки

Зробивши аналіз для створення даного чат-бота було виявлено, що для написання такого бота необхідно використовувати одну з мов серверного програмування: Python, Node.JS, PHP. Треба розуміти та визначити яка саме мова найкраще підходить для розробки такого бота. Насамперед є важливим вміння працювати з REST (Representational State Transfer) API (Application Programming Interface), та їх аналогами які надають месенджери, а в даному випадку це – Telegram Bot API. Після цього важливо визначитися з типом бота:
а) Перший тип ботів, що мають здібність до навчання (тобто розуміють природні мови), які вміють використовувати логіку при спілкуванні з користувачем та обробляти природну мову для створення відповіді на повідомлення;

б) Другий тип заскриптовані боти (боти, які не здатні зрозуміти природню мову), у них діалог з користувачем це заздалегідь сформований набір слів або речень, а «скрипт» це своєрідне дерево рішень, в якому сценарій, який був запрограмований заздалегідь є відповіддю на питання користувача. Діалоги в них в більшості випадків лінійні і структуровані. І на останок, потрібно визначити мету бота, оскільки у нього повинна бути певна мета, тому що в іншому випадку він не матиме сенсу.

Для себе я обрав другий тип ботів.

Також для цього бота потребується робота з базою , для опрацювання ставок та зберігання інформації о користувачах.

Для цього я вибрав раніше знайому для мене БД MYSQL , там є веб інтерфейс для інтуїтивно зрозумілих кроків створення і налаштування таблицями и самою базою даних.

Python – це мова програмування загального призначення, яка націлена перш за все на збільшення продуктивності самого розробника ПЗ, ніж коду, який він пише. Якщо говорити простою мовою, Python дає можливість написати практично все: веб- / настільні додатки, ігри, автоматизовані інформаційні системи, комплексні системи, системи управління життєзабезпеченням і багато іншого. Крім того, поріг входження низький, а код багато в чому небагатослівний і зрозумілий навіть для того, хто ніколи на ньому не писав. За рахунок простоти коду, майбутній супровід програм, які були написані на Python, стає легше і приємніше по відношенню до Java або C. А з точки зору бізнесу це дає скорочення витрат і підвищення ефективності трудових ресурсів. Практично на всіх платформах і операційних системах був реалізований інтерпретатор Python, що є безперечною перевагою. Першою такою мовою був C, однак його типи даних на різних машинах досить часто займали різну кількість пам'яті, а це представляє деяку перешкоду при написанні великих програм. Також, важлива риса – розширюваність мови, цьому надається велике значення і, як пише сам автор Гвідо ван Россум, мова була задуманий саме як

розширювана. Це означає, що є можливість вдосконалення мови усіма зацікавленими розробниками. Інтерпретатор був розроблений на мові програмування С і вихідний код доступний для будь-яких змін. У разі необхідності, можна вставити його в свою програму і використовувати як власну вбудовану оболонку. Або ж, написавши на С доповнення до Python і скомпілювавши програму, отримати "розширений" інтерпретатор з новими можливостями.

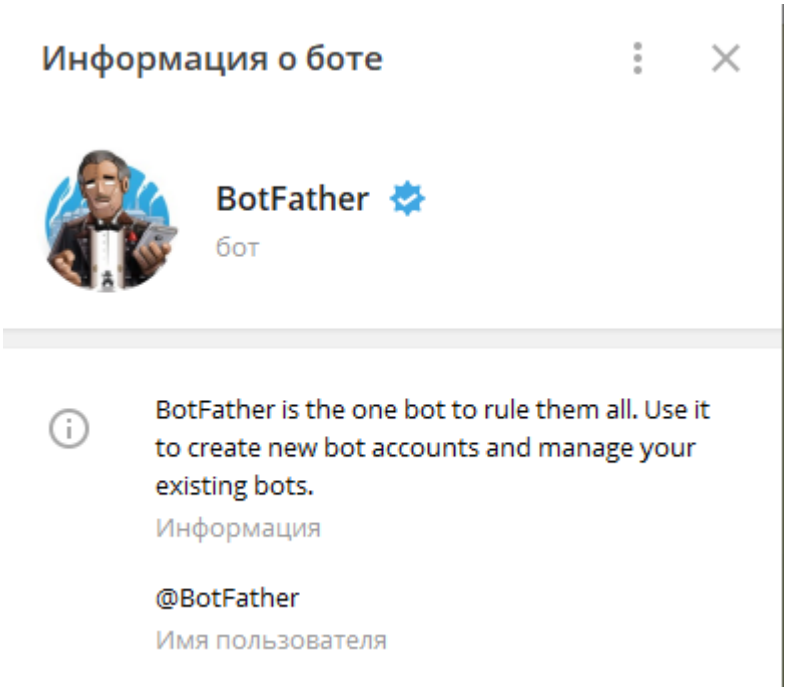
Модуль TeleBot – є оболонкою над запитами до TelegramBotAPI, використовується для спрощення і мінімізації написаного коду. Всі типи визначені в types.py. Всі вони повністю відповідають визначенням типів API Telegram, за винятком from поля Message, яке перейменовано в from_user (оскільки from це зарезервований токен Python). До таких атрибутів, як message_id можна звертатися безпосередньо, наприклад: message.message_id. При написанні програм не дуже вимогливих до швидкості виконання варто звернути увагу, що атрибут message.chat може належати як певному користувачеві, так і до групового чату, це з повністю окупається перевагами мови.

В класі TeleBot розташовані всі методи API. Щоб слідувати загальним угодам про імена Python, вони перейменовані. Наприклад: sendMessage send_message, editMessageText ,edit_message_text. Функція, прикрашена декоратором примірника TeleBot – обробник повідомлень. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає True або False для певного повідомлення і обробник отримує дозвіл на обробку повідомлення, якщо повертається True.

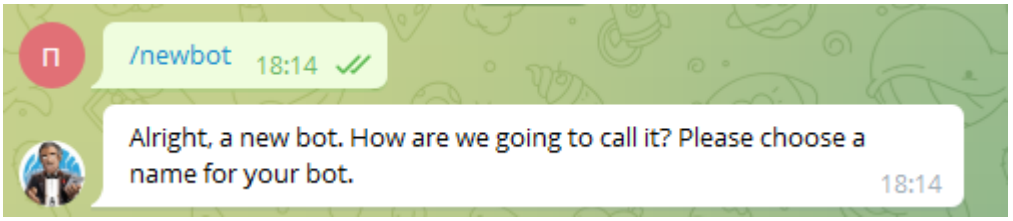
РОЗДІЛ 2. Створення Telegram-bot та отримання API Token

2.1 Отримання API Token

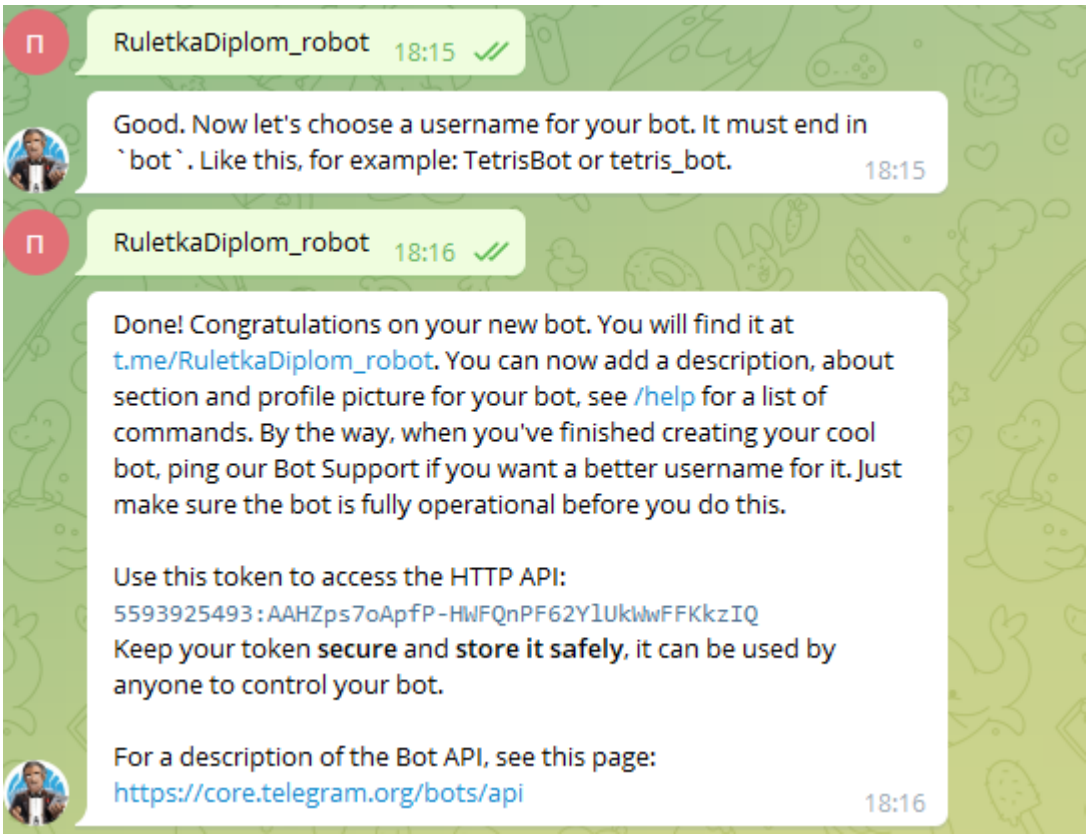
Бот взаємодіє з користувачами за допомогою Bot API. Керування ботами на платформі доступне в боті BotFather. Зокрема, там можна створити бота та переглянути чи змінити присвоєний йому токен API. Пошуком в телеграмі знаходимо @BotFather та запускаємо його в чаті.



Для створення боту треба написати команду /newbot.



Далі треба вибрати ім'я боту, наступний пункт Нік боту , обов'язково треба щоб в ньому було слово ‘bot’ . Якщо все добре , вам дасть повідомлення з посиланням на бота та його API Token.



Для нас потрібен ключ токен, за допомогою його ми маємо повний контроль над ботом.

```
5593925493:AAHZps7oApfP-HWFQnPF62YlUkWwFFKkzIQ
```

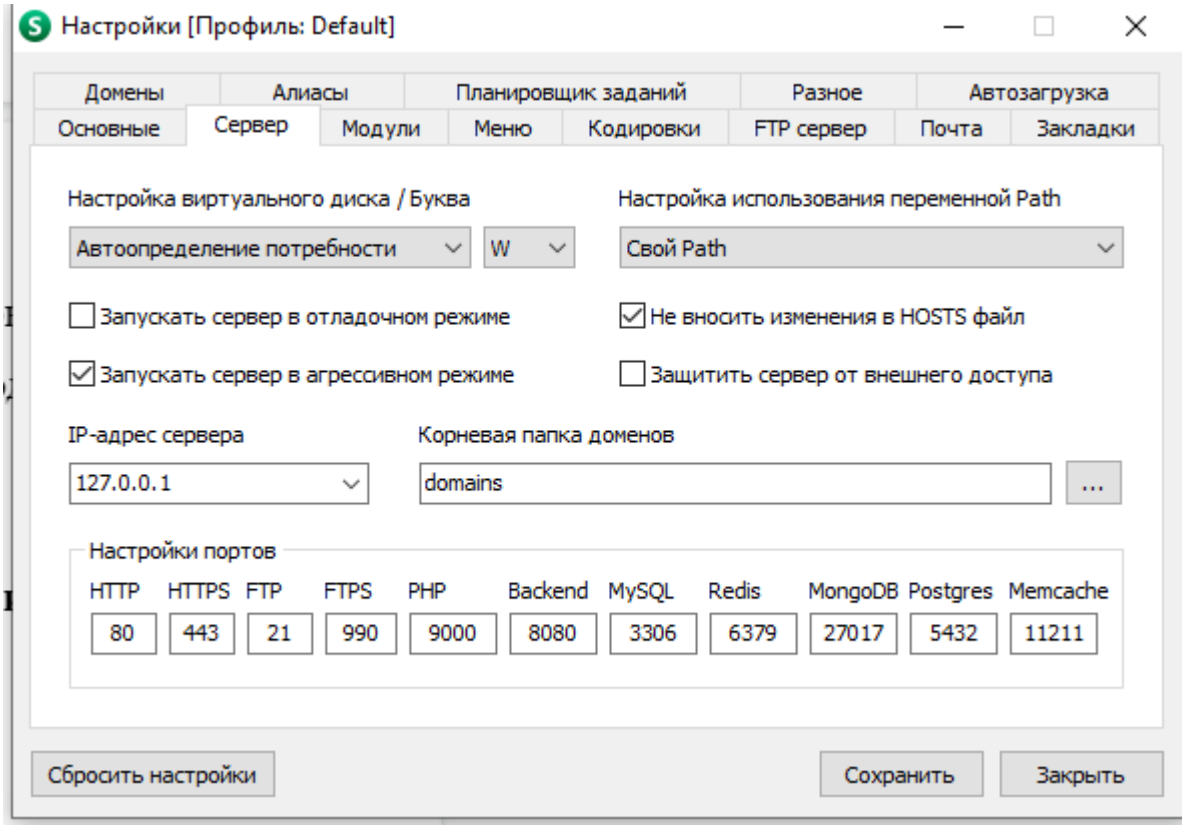
На цьому кроку ми закінчуємо роботу с Botfather, але якщо є бажання то можно зробити налаштування оформлення своего бота(ім'я , фото , інформація, команди).

РОЗДІЛ 3. Створення бази даних MYSQL

3.1 Піднімання серверу на localhost

Я буду використовувати готовий сервер OSPANEL. Там вже преінстальоване база даних MYSQL та веб-інтерфейс для неї PhpMyAdmin(для зручної орієнтації та створення таблиць в базі)

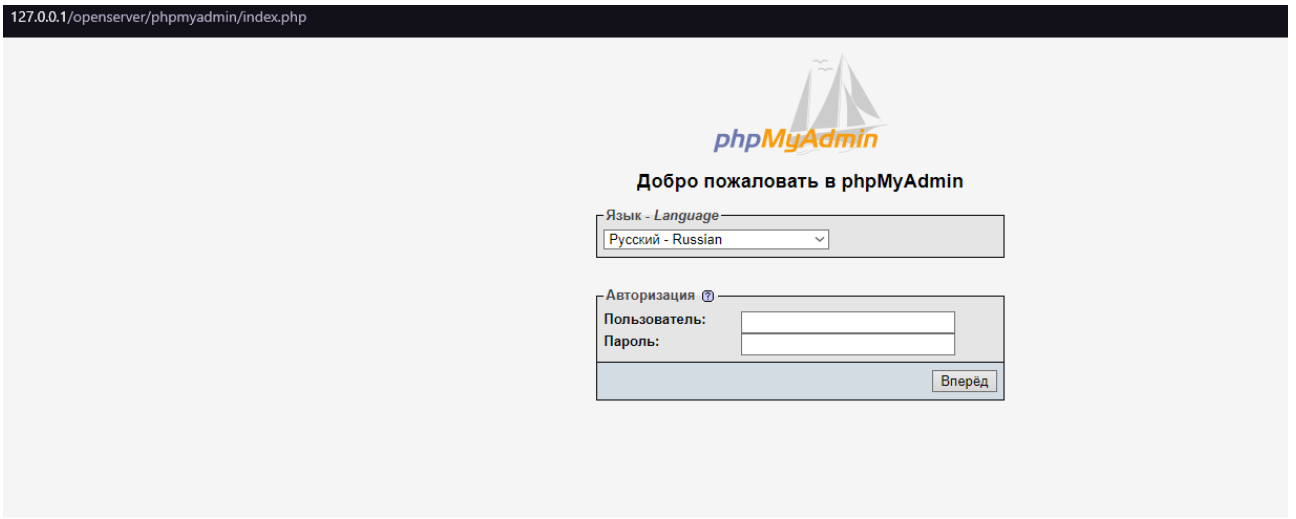
.Також можно використовувати MYSQL менеджер або консоль , кому як зручніше Після інсталювання серверу , запускаю його для подальшої роботи с базою. Так виглядає головне меню налаштувань сервера.



3.2 Вхід в панель PhpMyAdmin та створення бази даних

Скориставшися адрессою:

http://127.0.0.1/openserver/phpmyadmin/index.php



ми потрапляємо на адмін панель та налаштуванням бази. Плюси цього web-інтерфейсу , що ми маємо можливість створювати базу даних та таблиці не використовуючи SQL запити , це економить час та зручність, але SQL запити тут також можна роботи. Інтерфейс інтуїтивний , тому легко розібратися. Створюємо базу даних для нашого телеграм боту. Спочатку добавляємо 4 таблиці: users,bank,payment,payout.

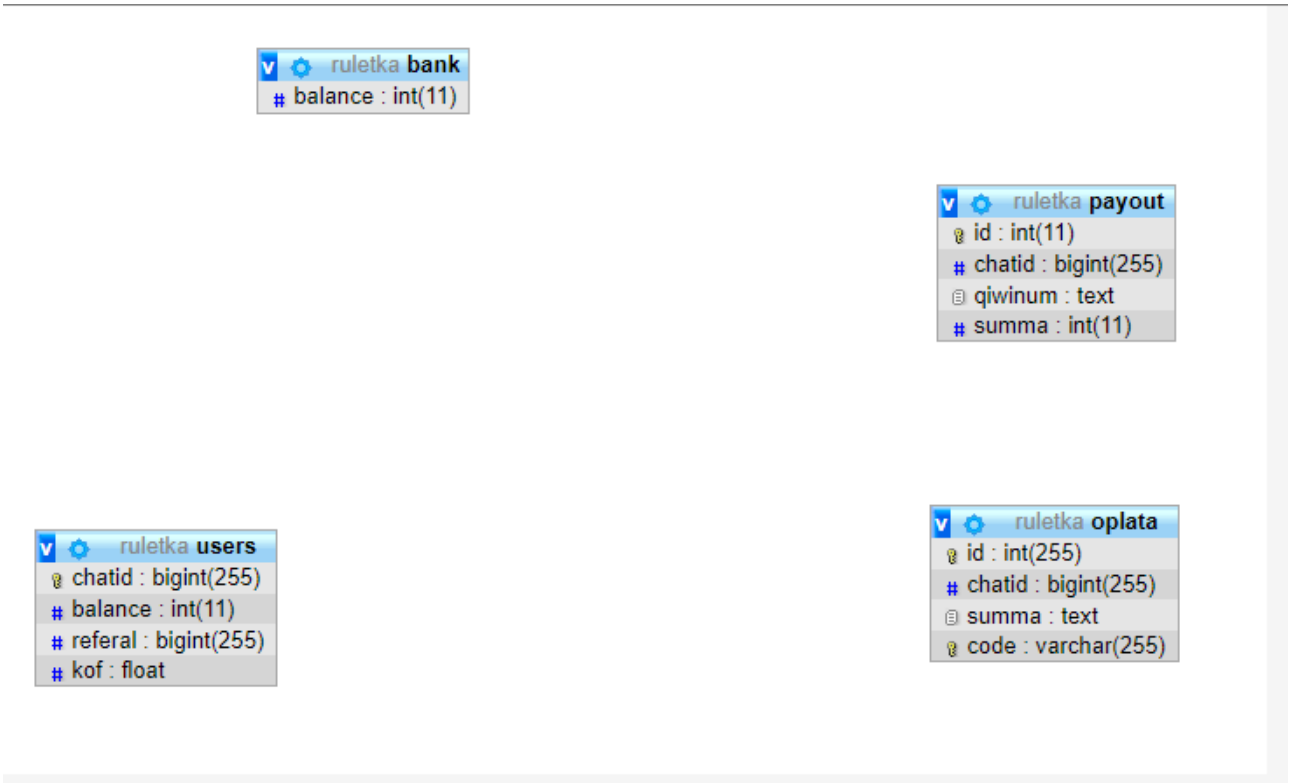
Users - таблиця для роботи з інформацією про користувача(Поля: chatid,balance, referal, kof)

Bank - Таблиця для роботи з інформації з балансом саомої рулетки.

Payment - Таблиця для роботи з платіжною системою, для поповнення рахунку. (id,chatid, amount, code)

Payout - Таблиця для роботи з платіжною системою, для виводу балансу.(id,chatid,numberwallet,amount)

Структура БД виглядає так.



Таблиця USERS виконує основну роль в зберіганні та обробки даних пов'язаних с користувачем.

Chatid - поле для ідентифікації юзера (є унікальним для кожного) використовується для звернення в кожному запиті.

Balance - поле для зберігання даних балансу користувача.

Referal - поле для зберігання Chatid реферала юзера.

kof - поле для зберігання коефіцієнта реферального бонусу

Таблиця BANK використовується для зберігання балансу банка рулетки. Визивається в деяких функціях для системи гри.

Balance - поле для зберігання балансу.

Таблиця PAYOUT виконує роль для історії виводу балансу юзерів.

Id- поле унікальне для ідентифікації транзакції

chatid - унікальне поле chatid юзера який виконує транзакцію на вивід балансу.

qiwinum - номер електронного гаманця куди проходить вивід.

summa - сума для виводу.

Таблиця OPLATA(PAYMENT) - використовується для перевірки оплати на баланс рулетки.

id- унікальне поле для ідентифікації платіжа

chatid - унікальне поле chatid юзера

summa - сума поповнення балансу

code - коментар для аутефікації транзакції

РОЗДІЛ 4. Створення платіжної системи

4.1 Отримання API Key гаманця

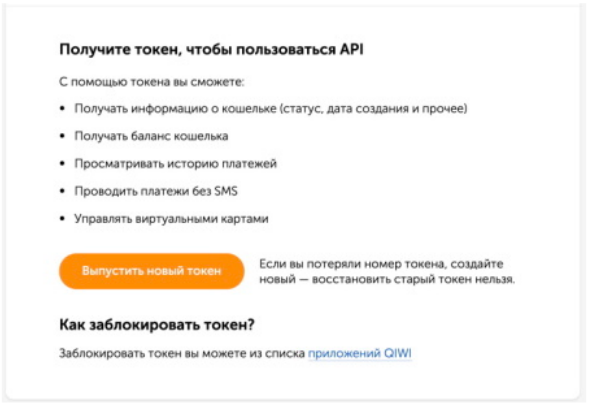
Отримання API key не є тяжкою задачею, треба мати лише електроний гаманець в системи QIWI та перейти в підрозділ API. Після отримання ключа ми можемо використовувати QIWI API в повній мірі та реалізувати платіжну систему.

Authorization data

QIWI Wallet API implements OAuth 2.0 open authorization protocol specification. A user registers or authenticates on https://qiwi.com QIWI Wallet site and requests a token with a certain scopes. Token issue is confirmed by SMS code.

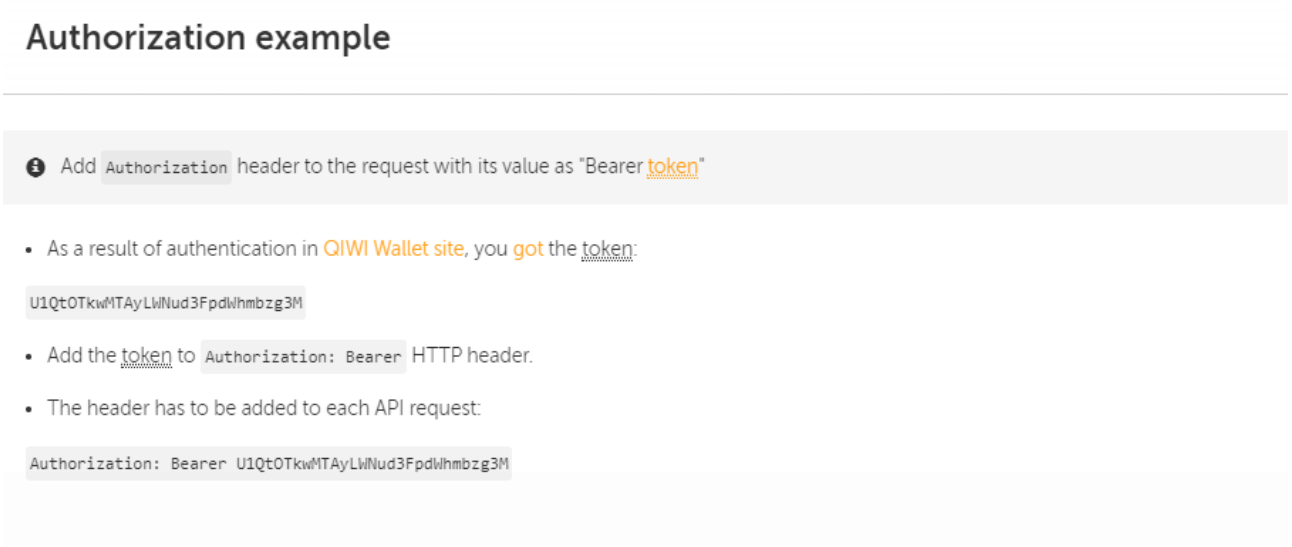
How to get a token

1. Open https://qiwi.com/api page in your browser. You will need to register or authenticate on QIWI Wallet service. Click on **Выпустить новый токен** (please note, interface is Russian only).



2. Select token scopes in the pop-up window and click **Продолжить**:
   - Запрос информации о профиле кошелька - allows use of person's profile requests, identification API, limits API
   - Запрос баланса кошелька - allows balance requests
   - Просмотр истории платежей - allows payments history requests
   - Проведение платежей без SMS - allows making payment requests with no SMS confirmation (regardless the settings https://qiwi.com/settings/options/security.action), using invoicing's API and notification service

Як тільки ми маємо ключ , можно реалізовувати за допомогою документації функції з нашим балансом. Ще один приклад як авторизуватися за допомогою ключа



4.2 Реалізація основних методів

Для роботи боту та всіх операцій нам потребується декілька функцій платіжної системи:

- Поповнення рахунку
- Перевірка балансу рахунку
- Надсилання платежій (вивід балансу)

Скориставшися документацією API починається розробка цих функцій.

4.3 Поповнення рахунку

Функція буде приймати об'єкт повідомлення де буде зберігатися дані о користувачі та саме повідомлення. Для початку треба підключитися к базі даних с допомогою модуля на python ‘mysql.connector’ це робимо. Потім генеруємо рандомне число для ініціалізації платежу. Отриманні дані - ід користувача, сума транзакції, рандомне число. Робимо SQL запит до бази даних та вносимо їх.

INSERT INTO payment(chatid,amount,code) VALUES('{}’, '{}','{}')

Закриваємо підключення к базі даних. Далі формуємо POST запрос із документації API для поповнення рахунку гаманця та передаємо користувачу.

Приклад POST запросу з документації для надсилання коштів

```
POST /sinap/api/v2/terms/99/payments HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Bearer YUu2qw048gtdsvlk3iu
Host: edge.qiwi.com

{
 "id":"11111111111111",
 "sum": {
  "amount":100.50,
  "currency":"643"
 },
 "paymentMethod": {
  "type":"Account",
  "accountId":"643"
 },
 "comment":"Комментарий",
 "fields": {
        "account":"+79121112233"
 }
}
```

4.4 Перевірка балансу рахунку

4.4.1 Перевірка поповнення

Для перевірки зарахування балансу який поповнив користувач. Та відображені в самому боті. Функція буде приймати об'єкт повідомлення де буде зберігатися дані о користувачі та саме повідомлення. Використовуючи API платіжної системи ми робимо GET запрос , щоб отримати дані з історії платежів нашого рахунку , зберігаємо ці дані. Далі підключаємо до бази даних , як ми робили це раніше та робимо запит до бази щоб отримати дані платежів конкретного користувача.

SELECT * FROM payment WHERE chatid = '{}' AND code = '{}'

Отримавши ці дані ми можемо порівняти з даними які отримали с бази та історію платежів. С допомогою циклу ми проходимо по всім платежам , які взяли раніше через API , якщо є збіг по сумі транзакції та унікальному коду, то ми зараховуємо на баланс користувача цю суму та видаляємо з бази цю транзакцію. І даємо повідомлення про поточний баланс.

Документація с API

```python
import requests

# История платежей - сумма за диапазон дат
def payment_history_summ_dates(my_login, api_access_token, start_Date, end_Date):
    s = requests.Session()
    s.headers['authorization'] = 'Bearer ' + api_access_token
    parameters = {'startDate': start_Date,'endDate': end_Date}
    h = s.get('https://edge.qiwi.com/payment-history/v2/persons/' + my_login + '/payments/total', params = parameters)
    return h.json()
```

4.4.2 Перевірка загального балансу гри (банку)

Функція будет отримувати на вхід логин гаманця та api key. Скориставшись GET запитом с документації ми повертаємо баланс банку(тобто весь баланс гаманця).

```python
bal = getbalancebank(QIWI_ACCOUNT, QIWI_TOKEN)['accounts']
bal1 = 0
for x in bal:
    bal1 = int(x['balance']['amount'])
```

4.5 Надсилання платежій (вивід рахунку)

Функція буде приймати об'єкт повідомлення де буде зберігатися дані о користувачі та саме повідомлення і номер гаманця користувача з сумою. Якщо сума виводу більша ніж баланс користувача то дає повідомлення про це. Використовуючи API документацію робимо POST запит для відправки коштів на заданий гаманець користувача та заданої суми. Якщо запит повертає успішний результат , то віднімаємо суму виводу від балансу користувача. Підключаемся к базі даних та оновлюємо баланс через SQL запит.
UPDATE users SET balance = '{}' WHERE users.chatid = '{}';
Закриваємо підключення та повідомляємо ,що операція успішна.

РОЗДІЛ 5. Написання коду та реалізація інтерфейсу на python

5.1 Реєстрація користувача

Коли гравець запускає бота , йде ідентифікація та реєстрація. Якщо комнда /start була запущена с деяким ключем, то система записує цей ключ як реферальний. Перевірка цього ключа на валідність. Підключення до бази та створення SQL запиту реєстрації

INSERT INTO users(chatid,balance,referal,kof)"

"VALUES('{}','{}','{}','0.4')

Та повідомлення , якщо все успішно.

5.2 Реферальна система

Реферальна система реалізована в одном із пунктів меню. В цьому блоці доступні такі кнопки: Дізнатися свій реферальний код, Оновити реферал, Статистика.

5.2.1 Дізнатися реферальний код

Функція дуже проста, вона лише повідомляє користувача його реферальий код. referal = https://t.me/X?start=Y

X - назва боту

Y - його унікальний chatid отриманий від телеграм.

5.2.2 Оновити реферал

Функція отримує від користувача реферальний код того хто його запросив. Перевіряється валідність реферального коду. Підключаемся к базі та робимо запит

SQL "UPDATE users SET referal = '{}' WHERE users.chatid = '{}';

Закриваємо підключення та повідомляємо користувача про успіх.

5.2.3 Статистика

Ця функція повідомляє користувача про його статистику рефералів. Підключаемся к базі робимо SQL запит
SELECT COUNT(*) from users where referal='{}';

Цим самим отримуємо кількість юзерів які мають цей реферальний код.
Далі за допомогою функція getkof() отримуємо коефіцієнт. Коефіцієнт - відсоткова ставка яку отримує користувач в разі програшу його рефералів. (За умовчанням це 40%) . Останнім кроком виводимо повідомлення з кількістю рефералів та його відсотковою ставкою.

5.3 Допоміжні функції

5.3.1 getkof() Отримання коефіцієнта користувача

Функція отримує унікальний чат айді користувача. Підключається к базі та робить SQL запит

SELECT kof FROM users WHERE chatid = '{}';

та повертає коефіцієнт .

5.3.2 getbalance() Отримання балансу користувача

Функція отримує унікальний чат айді користувача. Підключається к базі та робить SQL запит

SELECT balance FROM users WHERE chatid = '{}';

та повертає дані.

5.3.3 getreferal() Отримання балансу користувача

Функція отримує унікальний чат айді користувача. Підключається к базі та робить SQL запит

SELECT referal FROM users WHERE chatid = '{}';

та повертає дані.

5.3.4 getbank(): Отримання балансу банку гри

Використовуючи API документацію робимо GET запит та отримуємо баланс електронного гаманця. Підключаемся к базі та робимо SQL запит

UPDATE bank SET balance='{}' WHERE 1

SELECT * FROM `bank`

та повертаємо дані про баланс.

5.4 Основні функції алгоритму гри.

5.4.1 Червоне

Якщо користувач вибирає червоне починається гра.

Створюємо список чисел які відносяться до червоних.

red= [32,19,21,25,34,27,36,30,23,5,16,1,14,9,18,7,12,3]

Генеруємо випадкове число в межах від 0 до 36.

Надсилаємо користувачу відео ,де показано результат випадкового числа на прикладі рулетки.

Далі перевіряємо якщо випадкове число належить нашому списку червоних,то оновлюємо баланс користувачу(додаємо суму ставки).

Робимо запит до бази та оновлюємо баланс

UPDATE users SET balance = '{}' WHERE users.chatid = '{}';

Повідомляємо про результати гри.

В іншому випадку , якщо випадкове число не належить списку червоних.

Робимо аналогічні операції , тільки навпаки. Віднімаємо баланс та записуємо в базу. Але в нас ще ж є реферальна система, коли гравець програє , 40% отримує та людина яка його запросила. За допомогою функції описаної раніше, визначаємо реферал гравця який програв(це є унікальний айді чата). Потім отримаємо баланс та коефіцієнт реферала. Та оновлюємо баланс за формулою balanceref = balanceref +(summ*kof) та добавляємо до бази.

5.4.2 Чорне

Функція чорне ідентична до такої ж функції червоне. Тому немає сенсу її описувати.

5.4.3 Парне

Якщо користувач вибирає парне починається гра.

Створюємо список чисел які відносяться до парне .

chetnoe = [2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36]

Генеруємо випадкове число в межах від 0 до 36.

Надсилаємо користувачу відео ,де показано результат випадкового числа на прикладі рулетки.

Далі перевіряємо якщо випадкове число належить нашому списку парне ,то оновлюємо баланс користувачу(добавляємо суму ставки).

Робимо запит до бази та оновлюємо баланс

UPDATE users SET balance = '{}' WHERE users.chatid = '{}';

Повідомляємо про результати гри.

В іншому випадку , якщо випадкове число не належить списку парних.

Робимо аналогічні операції , тільки навпаки. Віднімаємо баланс та записуємо в базу. Але в нас ще ж є реферальна система, коли гравець програє , 40% отримує та людина яка його запросила. За допомогою функції описаної раніше, визначаємо реферал гравця який програв(це є унікальний айді чата). Потім отримаємо баланс та коефіцієнт реферала. Та оновлюємо баланс за формулою balanceref = balanceref +(summ*kof) та добавляємо до бази.

5.4.4 Непарне

Функція непарне ідентична до такої ж функції парне. Тому немає сенсу її описувати.

5.4.5 Дюжина

Користувач вибирає дюжину для гри (1 , 2 , 3) починається гра.

Створюємо список чисел які відносяться до 3-х дюжин.

duzina1= [1,2,3,4,5,6,7,8,9,10,11,12]

duzina2 = [13,14,15,16,17,18,19,20,21,22,23,24]

duzina3= [25,26,27,28,29,30,31,32,33,34,35,36]

Генеруємо випадкове число в межах від 0 до 36.

Надсилаємо користувачу відео ,де показано результат випадкового числа на прикладі рулетки.

Далі перевіряємо якщо випадкове число належить нашому списку дюжини, яку він вибрав,то

оновлюємо баланс користувачу(добавляємо суму ставки balance = balance + (summ * 2)).

Робимо запит до бази та оновлюємо баланс

UPDATE users SET balance = '{}' WHERE users.chatid = '{}';

Повідомляємо про результати гри.

В іншому випадку , якщо випадкове число не належить списку дюжини.

Робимо аналогічні операції , тільки навпаки. Віднімаємо баланс та записуємо в базу. Але в нас ще ж є реферальна система, коли гравець програє , 40% отримує та людина яка його запросила. За допомогою функції описаної раніше, визначаємо реферал гравця який програв(це є унікальний айді чата). Потім отримаємо баланс та коефіцієнт реферала. Та оновлюємо баланс за формулою balanceref = balanceref +(summ*kof) та добавляємо до бази.

5.4.6 Число

Користувач вибирає число для гри, починається гра.

Створюємо список чисел.

chisla =[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36]

Генеруємо випадкове число в межах від 0 до 36.

Надсилаємо користувачу відео ,де показано результат випадкового числа на прикладі рулетки.

Далі перевіряємо якщо випадкове число належить нашому списку, яку він вибрав,то

оновлюємо баланс користувачу(добавляємо суму ставки balance = balance+(summ*35)

)).

Робимо запит до бази та оновлюємо баланс

UPDATE users SET balance = '{}' WHERE users.chatid = '{}';

Повідомляємо про результати гри.

В іншому випадку , якщо випадкове число не належить списку чисел.

Робимо аналогічні операції , тільки навпаки. Віднімаємо баланс та записуємо в базу. Але в нас ще ж є реферальна система, коли гравець програє , 40% отримує та людина яка його запросила. За допомогою функції описаної раніше, визначаємо реферал гравця який програв(це є унікальний айді чата). Потім отримаємо баланс та коефіцієнт реферала. Та оновлюємо баланс за формулою balanceref = balanceref +(summ*kof) та добавляємо до бази.

5.5 Інтерфейс бота

Застосовуючи Telegram API , створюємо меню для бота.

Меню складається:

1. Головне меню
   1.1. Ставки
      1.1.1. Вхідні дані ставки

1.1.1.1. Червоне
1.1.1.2. Чорне
1.1.1.3. Число
1.1.1.4. Парні
1.1.1.5. Непарні
1.1.1.6. Дюжина
1.1.1.7. Відхилення ставки

1.2. Інформація

1.3. Баланс

1.3.1. Перевірка балансу
1.3.2. Поповнення балансу
1.3.3. Вивід грошей
1.3.4. Назад

1.4. Реферальна Система

1.4.1. Дізнатися свій реферальний код
1.4.2. Оновити реферал
1.4.3. Статистика
1.4.4. Назад

За допомогою декоратора ми обробляємо дії користувача. Головна функція яка відповідає за обробку текста відправленим юзером до бота.
Принцип її такий. що якщо користувач відправляє текст до бота, то спрацьовує цей декоратор. Далі в головній функції йде обробка цього тексту , якщо текст належить пунктам із головного меню то код виконується , інакше нічого не відбувається.

При вводу тексту “Ставки” , виконується наступна логіка. Дізнатися баланс користувача за допомоги допоміжної функції , та надсилає повідомлення та запит на ввід суми ставки. Далі перехід до іншої функції перевірки суми на

валідність . А саме перевіряється поточний баланс с сумою ставки , якщо сума ставки менша за баланс , то повідомляє про це користувача. В Іншому випадку повідомляє ,що ставка була поставлена. Далі вже виконується функції основні , такі як Червоне/Чорне , Парне/Непарне, Число, Дюжини. Кнопка відхилення ставки , повертає користувача до головного меню.

При вводу тексту “Інформація” , користувачу надається повідомлення про бота та основну інформацію.

При вводу тексту “Баланс” , нас перекидає на наступний блок меню.
Всі функції балансу були описані раніше.

При вводу тексту “Реферальна система” , нас перекидає на наступний блок меню.
Всі функції реферальної системи були описані раніше.

ВИСНОВКИ

В рамках дипломної роботи був розроблений телеграм бот онлайн-рулетка.

Оптимізація та зручність роботи. В даний час більшість людей спілкуються в месенджерах, тому це гарна платформа для розробки. Telegram BOT є гарний аналог будь-якого сайту, тому що можно реалізувати будь-який функціонал , а це є і дешевше і зручніше для користувачів. Тому актуальність даної задачі дуже велика.

Я ознайомився з технологіями Telegram API, MYSQL, QIWI API , Python.

В результаті встановлення необхідних компонентів та створення програми, були отримані результаті, що були очікуваними. Результати, що були отримані в результаті курсової роботи далекі від сучасних досягнень та стандартів, але розробку ботів треба поширювати та вдосконалювати.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. [Електронний ресурс] **https://core.telegram.org/bots/api**
2. [Електронний ресурс] **https://t.me/BotFather**
3. [Електронний ресурс] https://uk.wikipedia.org/wiki/Прикладний_програмний_інтерфейс
4. [Електронний ресурс] https://uk.wikipedia.org/wiki/Telegram
5. [Електронний ресурс] https://qiwi.com/api
6. [Електронний ресурс] https://ospanel.io
7. [Електронний ресурс] https://stackoverflow.com
8. [Електронний ресурс] https://www.google.com