**Інформаційні мережі та телекомінікації**

**Лабораторна робота №5.**

**Тема:** Інтеграція зовнішніх систем у власний telegram bot.

**Мета:** Навчитися використовувати зовнішні системи під час взаємодії з користувачем.

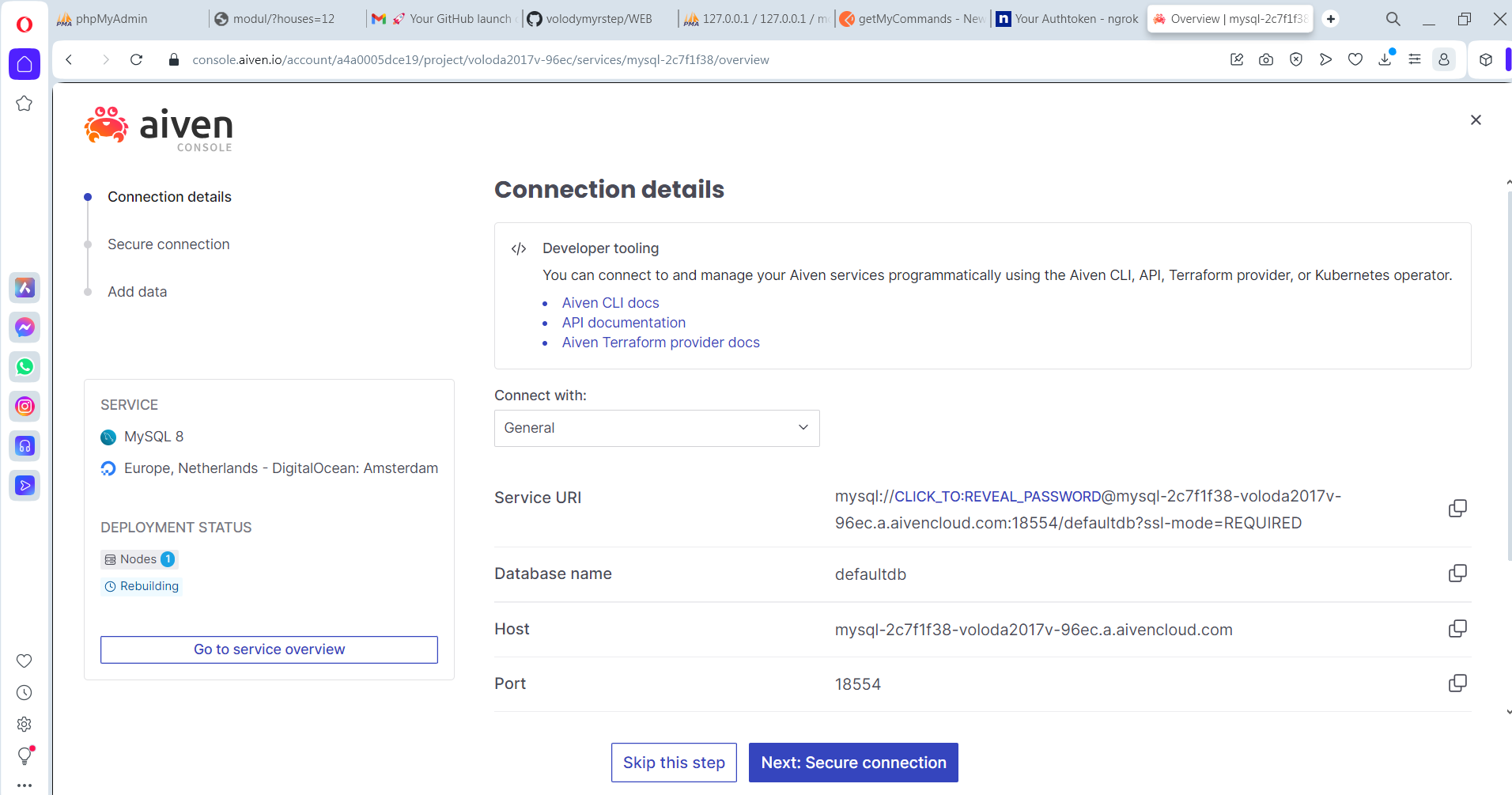
**Хiд роботи**

**І. Виконайте наступні завдання** (у файлі звіту опишіть відповідні процеси):

Продовжимо розробку telegram bot на python. В цій лабораторній підключимо зовнішню базу даних mysql для збереження даних від користувача. Представимо, що нам необхідно створити базу даних користувачів магазину і через бот зробити можливість реєстрації користувачів.

Результат виконання цієї лабораторної закомітьте в гілку lab5 вашого проекту.

**Завдання 1.** Зареєструватися на сайті <https://aiven.io/free-mysql-database> взявши безкоштовний план і створити базу даних shop.



Далі використовуючи sql manager, або microsoft sql management server, або phpMyAdmin або php adminer або напряму використовуючи пакет mysql в системі, підключіться до сервера бази даних (інформація про підключення в пункті меню Overview) та створіть таблицю customers, яка буде містити id(налаштуйте як primary key, autoincrement), Ім’я, Прізвище та номер телефону користувача, а також його telegram id (chat\_id)

Встановіть пакети для роботи з mysql з сторони python

**CREATE** **TABLE** customers (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

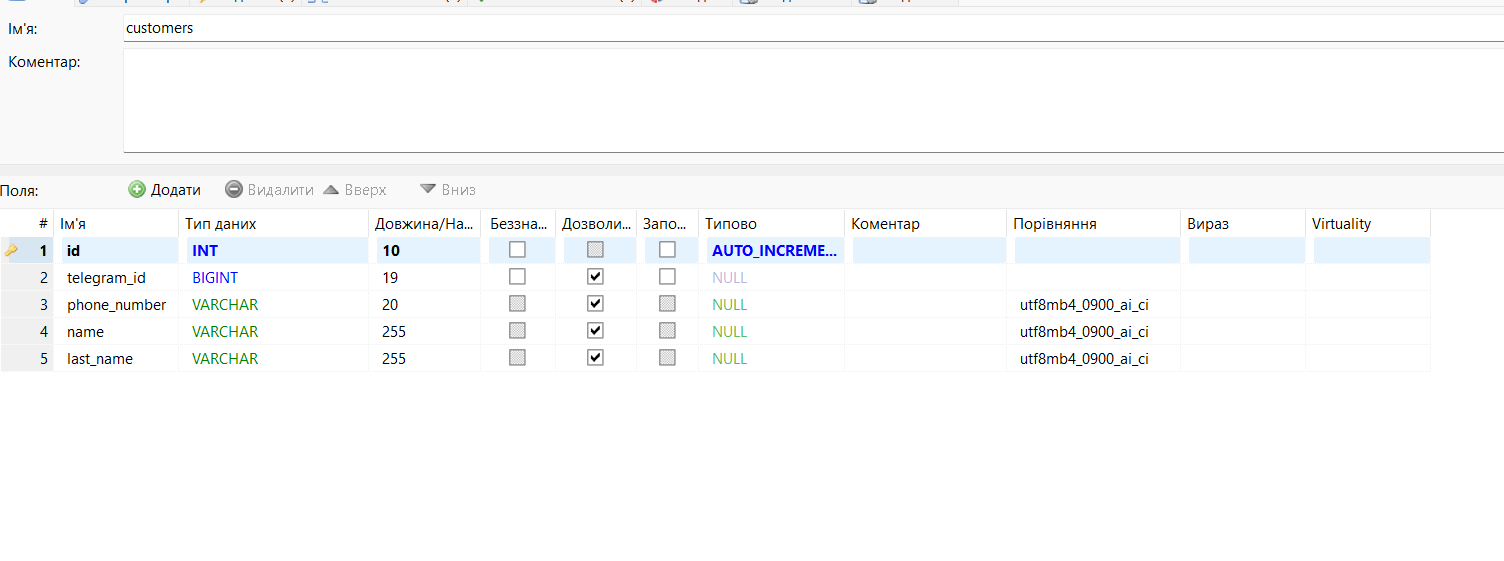
telegram\_id **BIGINT**,

phone\_number **VARCHAR**(20),

**name** **VARCHAR**(255),

last\_name **VARCHAR**(255)

);



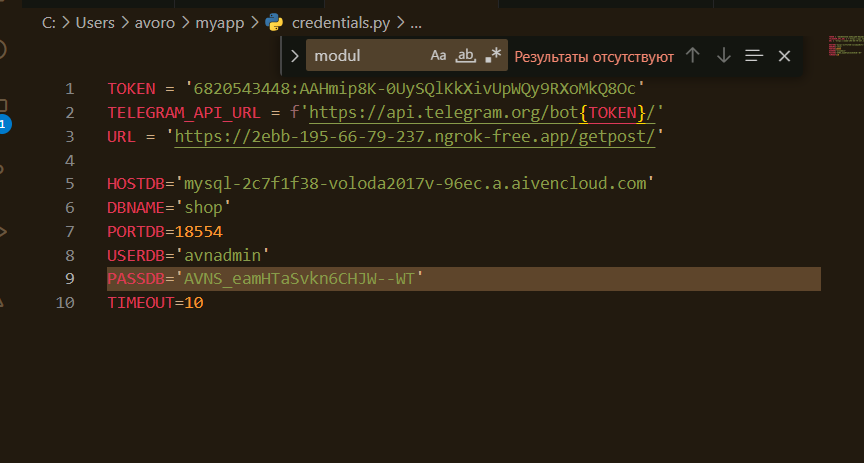
python -m pip install pymysql

python -m pip install cryptography

Додайте в credentials.py нові змінні для роботи з mysql

HOSTDB, DBNAME, PORTDB, USERDB, PASSDB, TIMEOUT

Значення візьміть з вкладки Overview на aiven



Приклад підключення до бази даних і взаємодії з mysql можете переглянути в цьому прикладі:

import pymysql

from myapp.credentials import HOSTDB, DBNAME, PORTDB, USERDB, PASSDB, TIMEOUT

timeout = TIMEOUT

connection = pymysql.connect(

charset="utf8mb4",

connect\_timeout=timeout,

cursorclass=pymysql.cursors.DictCursor,

db=DBNAME,

host=HOSTDB,

password=PASSDB,

read\_timeout=timeout,

port=PORTDB,

user=USERDB,

write\_timeout=timeout,

)

try:

cursor = connection.cursor()

cursor.execute("CREATE TABLE customers (id INTEGER PRIMARY KEY)")

cursor.execute("INSERT INTO customers (id) VALUES (1), (2)")

cursor.execute("SELECT \* FROM customers ")

print(cursor.fetchall())

finally:

connection.close()

**Завдання 2.** Створити нову команду /register в меню бота, додати описову інформацію про те, що дана команда виконує реєстрацію нового користувача. Ви це можете зробити двома способами:

* через postman використовуючи /setMyCommands
* або через наш python бот, зареєструвавши новий шлях, аналогічно як ми робити для /setwebhook



**Завдання 3.** Розробити функціонал команди /register бота таким чином, щоб користувач міг пошарити свої дані для створення нового профілю.

Ось приклад запиту для отримання об’єкту контакту користувача через кнопку контакту через команду /sendMessage:

{

"chat\_id": "ХХХХХХХХХХХХ",

"text": "Please send your contact:",

"reply\_markup": {

"keyboard": [

[

{

"text": "My phone",

"request\_contact": true

}

]

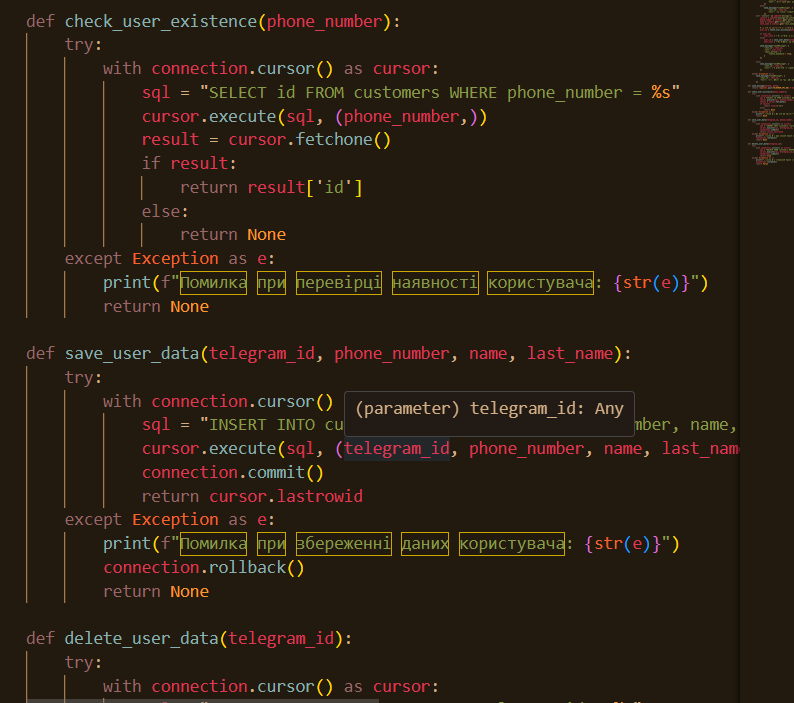
],

"resize\_keyboard": true,

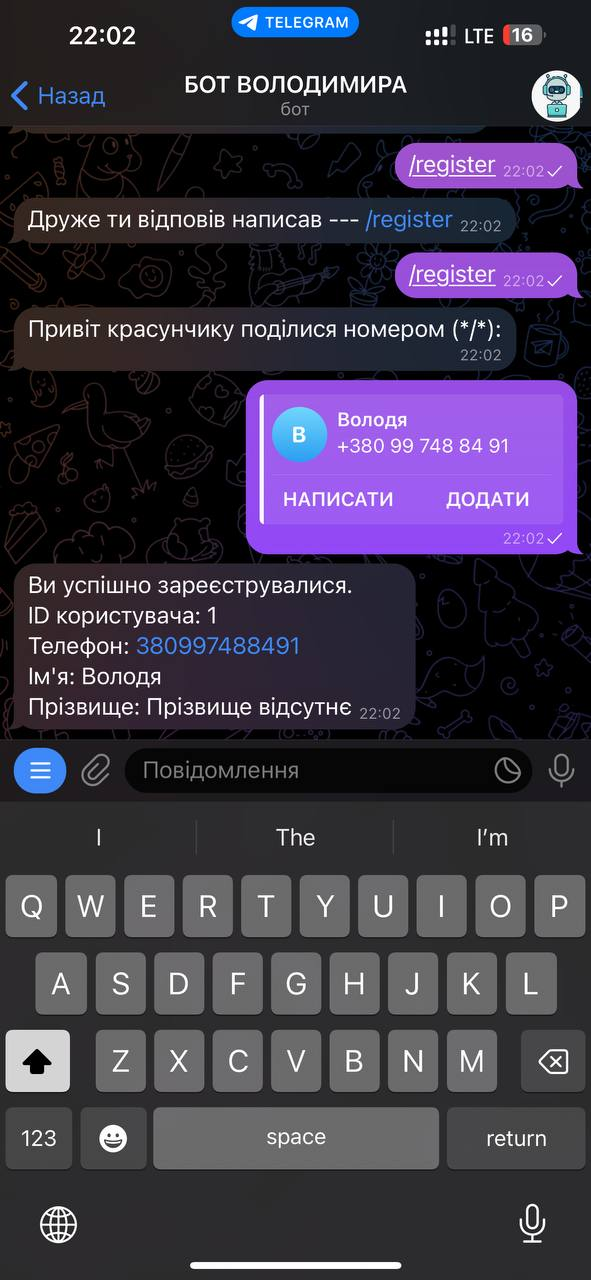
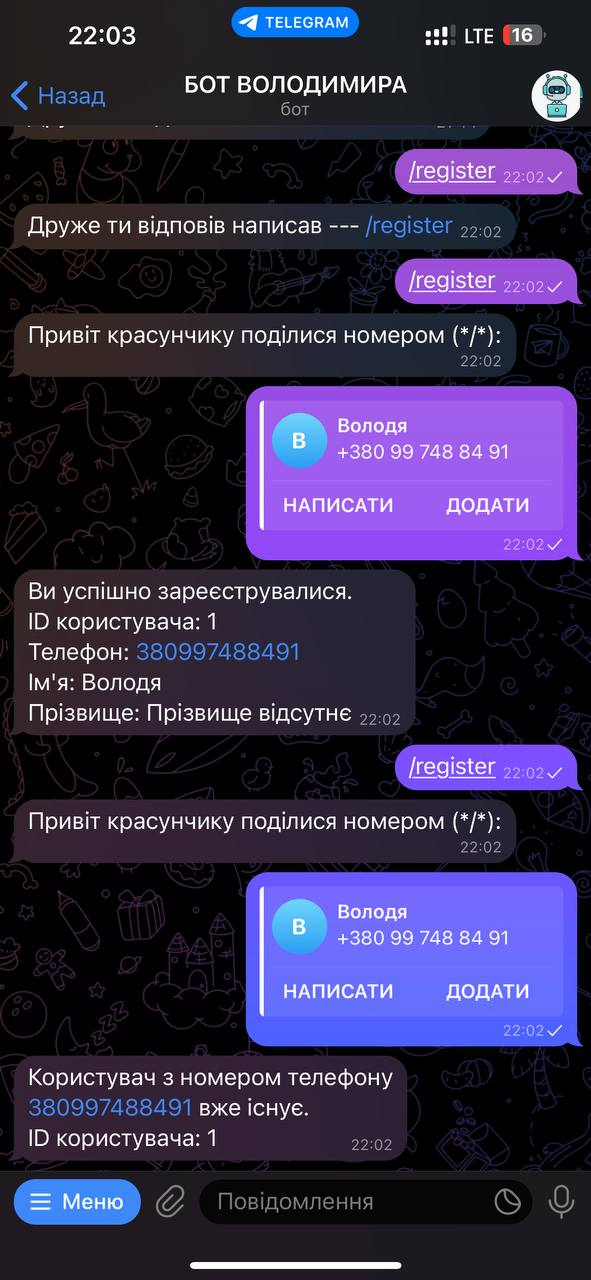
"one\_time\_keyboard": true

}

}



**Завдання 4.** Отримавши всі необхідні дані від користувача, перевірити за номером телефону, чи існує такий користувач в базі даних. Якщо існує, то повернути відповідь що користувач існує і його ІД.



**Завдання 5.** Якщо це новий користувач, зберегти дані в базу даних таблицю customers і повернути користувачу повідомлення про успішну реєстрацію та його ІД.

