

ДОДАТОК А

Програмний код

Додаток А.1 – Код файлу Bot.py

```

-*- coding: utf-8 -*-
import config
from sqlite import SQLite
import os
import asyncio
from tracking import check_tracking_number
import logging
from aiogram import Bot, Dispatcher, executor, types
from aiogram.dispatcher.filters import Text
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import State, StatesGroup
logging.basicConfig(level=logging.INFO)
storage = MemoryStorage()
bot = Bot(token=config.API_TOKEN)
dp = Dispatcher(bot, storage=storage)
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
database = os.path.join(BASE_DIR, "tracking.db")
db = SQLite(database)
class Form(StatesGroup):
    tracking_number = State()
    tracking_title = State()
class Tracking(StatesGroup):
    tracking_number = State()
    keyboard_markup = types.ReplyKeyboardMarkup(row_width=1,
        resize_keyboard=True)
    keyboard_markup.row(types.KeyboardButton("Перевіритис статус
    відправлення"))
    keyboard_markup.row(types.KeyboardButton("Збережені відправлення"))
    keyboard_markup_create = types.ReplyKeyboardMarkup(row_width=1,
        resize_keyboard=True)
    keyboard_markup_create.row(types.KeyboardButton("Створити"))
    keyboard_markup_create.row(types.KeyboardButton("Повернутися ←"))
    keyboard_markup_question = types.ReplyKeyboardMarkup(
        resize_keyboard=True, selective=True)
    keyboard_markup_question.add("Зберегти номер для подальших відстежань")
    def remove_html_tags(text):
        import re
        clean = re.compile('<.*?>')
        return re.sub(clean, '', text)
    @dp.message_handler(commands=['start'])
    async def send_welcome(message: types.Message):
        await message.answer("Вітаю! \n\nЦей бот призначений для відстеження
        поштових відправлень різних поштових
        сервісів",reply_markup=keyboard_markup)
    @dp.message_handler(state=None)
    async def all_msg_handler(message: types.Message):
        if message.text == 'Збережені відправлення':
            reply_text = "Завантаження збереженого списку..."
            await message.answer(reply_text, reply_markup=types.ReplyKeyboardRemove())
            tracking_list = db.get_all_tracking(message.from_user.id)
            await asyncio.sleep(2)
            if len(tracking_list) == 0:

```

```

reply_text = 'Список пустий. Якщо бажаєте створити новий запис натисніть
кнопку "Створити" нижче👇'
await message.answer(reply_text, reply_markup=keyboard_markup_create)
else:
reply_text = f'Список поштових відправлень'
await message.answer(reply_text)
for item in tracking_list:
id = item[0]
title = item[2]
number = item[3]
keyboard_markup_check_tracking = types.InlineKeyboardMarkup(row_width=2)
keyboard_markup_check_tracking.add(
types.InlineKeyboardButton('Перевірити статус',
callback_data=f"check_{id}"),
types.InlineKeyboardButton('Видалити', callback_data=f"delete_{id}"),
)
reply_text = f'Поштове відправлення \n{title} ({number})'
await message.answer(reply_text,
reply_markup=keyboard_markup_check_tracking)
reply_text = 'Якщо бажаєте створити новий запис натисніть кнопку
"Створити" нижче👇'
await message.answer(reply_text, reply_markup=keyboard_markup_create)
# await message.answer(reply_text, reply_markup=start_test_kb)
if message.text == "Перевірити статус відправлення":
await Tracking.tracking_number.set()
reply_text = "Введіть номер відправлення"
await message.answer(reply_text, reply_markup=types.ReplyKeyboardRemove())
if message.text == "Створити":
await Form.tracking_number.set()
reply_text = "Введіть номер поштового відправлення"
await message.answer(reply_text, reply_markup=types.ReplyKeyboardRemove())
if message.text == "Повернутися ←":
reply_text = "Оберіть дію нижче 👇"
await message.answer(reply_text, reply_markup=keyboard_markup)
@dp.callback_query_handler(Text(startswith="check_"))
async def check_tracking_callback(query: types.CallbackQuery):
current_id = int(query.data.split('_')[1])
current_item = db.get_tracking_info_by_id(current_id)
answer =
remove_html_tags(check_tracking_number(current_item[0][3]).split('Поштовий
сервіс')[0])
await query.answer(answer, show_alert=True)
@dp.callback_query_handler(Text(startswith="delete_"))
async def delete_tracking_callback(query: types.CallbackQuery):
current_id = int(query.data.split('_')[1])
current_item_message = db.delete_tracking_by_id(current_id)
await query.answer(current_item_message, show_alert=True)
reply_text = "Завантаження збереженого списку..."
await query.message.answer(reply_text,
reply_markup=types.ReplyKeyboardRemove())
tracking_list = db.get_all_tracking(query.from_user.id)
await asyncio.sleep(2)
if len(tracking_list) == 0:
reply_text = 'Список пустий. Якщо бажаєте створити новий запис натисніть
кнопку "Створити" нижче👇'
await query.message.answer(reply_text,
reply_markup=keyboard_markup_create)
else:
reply_text = f'Список поштових відправлень'
await query.message.answer(reply_text)

```

```

for item in tracking_list:
    id = item[0]
    title = item[2]
    number = item[3]
    keyboard_markup_check_tracking = types.InlineKeyboardMarkup(row_width=2)
    keyboard_markup_check_tracking.add(
        types.InlineKeyboardButton('Перевірити статус',
            callback_data=f"check_{id}"),
        types.InlineKeyboardButton('Видалити', callback_data=f"delete_{id}"),
    )
    reply_text = f'Поштове відправлення \n{title} ({number})'
    await query.message.answer(reply_text,
        reply_markup=keyboard_markup_check_tracking)
    reply_text = 'Якщо бажаєте створити новий запис натисніть кнопку
"Створити" нижче👇'
    await query.message.answer(reply_text,
        reply_markup=keyboard_markup_create)
    @dp.message_handler(state=Tracking.tracking_number)
    async def load_tracking_number(message: types.Message, state: FSMContext):
        async with state.proxy() as store:
            store['tracking_number'] = message.text
        async with state.proxy() as data:
            await message.answer("Пошук інформації 📄", parse_mode="html",
                reply_markup=types.ReplyKeyboardRemove())
            answer = check_tracking_number(data['tracking_number'])
            await asyncio.sleep(2)
            await message.answer(answer, parse_mode="html",
                reply_markup=keyboard_markup)
            await state.finish()
    @dp.message_handler(state=Form.tracking_number)
    async def load_form_tracking_number(message: types.Message, state:
        FSMContext):
        async with state.proxy() as store:
            store['tracking_number'] = message.text
        await Form.next()
        await message.answer("Введіть назву поштового відправлення")
    @dp.message_handler(state=Form.tracking_title)
    async def load_form_tracking_title(message: types.Message, state:
        FSMContext):
        async with state.proxy() as store:
            store['tracking_title'] = message.text
        async with state.proxy() as data:
            answer = db.add_tracking(message.from_user.id, data['tracking_title'],
                data['tracking_number'])
            await asyncio.sleep(2)
            await message.answer(answer, reply_markup=keyboard_markup_create)
        await state.finish()
    #
    if __name__ == '__main__':
        executor.start_polling(dp, skip_updates=True)

```

Додаток А.2 – Код файлу Config.py

```
API_TOKEN = '5671583701:AAHcTozJmZ4DD2L-ofWkRZT3z1QHLe7rh2M'
```

Додаток А.3 – Код файлу SQLite.py

```

import sqlite3
class SQLite:
def __init__(self, database):
self.connection = sqlite3.connect(database)
self.cursor = self.connection.cursor()
# def get_tracking_list(self):
# with self.connection:
# return self.cursor.execute("SELECT * FROM `tracking_items`").fetchall()
def add_tracking(self, user_id, tracking_title, tracking_number):
with self.connection:
self.cursor.execute("INSERT INTO `tracking_items` (`user_id`,
`tracking_number`, `tracking_title`) VALUES(?,?,?)", (user_id,
tracking_title, tracking_number))
return "Дані збережено!"
def get_all_tracking(self, user_id):
with self.connection:
return self.cursor.execute(f"SELECT * FROM `tracking_items` WHERE
`user_id` = {user_id}").fetchall()
def get_tracking_info_by_id(self, id):
with self.connection:
return self.cursor.execute(f"SELECT * FROM `tracking_items` WHERE `id` =
{id}").fetchall()
def delete_tracking_by_id(self, id):
with self.connection:
self.cursor.execute(f"DELETE FROM `tracking_items` WHERE `id` = {id}")
return "Запис видалено!"
def close(self):
"""Закриваємо з'єднання з БД"""
self.connection.close()

```

Додаток А.3 – Код файлу Tracking.py

```

import requests
import json
from bs4 import BeautifulSoup
from fake_useragent import UserAgent
import base64
from datetime import datetime
def check_tracking_number(track_id):
res_list = []
def charCodeAt(string, index):
if isinstance(string, str) and isinstance(index, int):
return ord(string[index])
# track_id = "59500000303585"
# track_id = "1Z2W218E6894138173"
# track_id = "4305909985"
for idex, num in enumerate(track_id):
res_list.append(charCodeAt(track_id, idex))
charCodeAt_result = ','.join(str(x) for x in res_list)
ua = UserAgent().random
urlAPI = f"https://api.trackposylka.com/parcels/"
headers = {
"accept": "application/json, text/javascript, */*; q=0.01",
"content-type": "application/x-www-form-urlencoded; charset=UTF-8",
"user-agent": ua,
}
session = requests.Session()

```

```

encoded_track_id = base64.b64encode(charCodeAt_result.encode("utf-8"))
r = session.post(urlAPI,data={'tracknumber': encoded_track_id,'service':
track_id},headers=headers)
bs = BeautifulSoup(r.text, 'lxml')
json_res = json.loads(bs.find('p').text)
message = f'Поштове відправлення № <b>{track_id}</b>\n'
if json_res["states"][-1].get("date"):
date = datetime.strptime(json_res["states"][-1]["date"], '%Y-%m-
%dT%H:%M:%S.%f%z').strftime('%d %B %Y')
time = datetime.strptime(json_res["states"][-1]["date"], '%Y-%m-
%dT%H:%M:%S.%f%z').strftime('%H:%I')
message += f'\n<b>Дата:</b>{date} о {time}'
if json_res["states"][-1].get("location"):
location = json_res["states"][-1]["location"]
message += f'\n<b>Місце знаходження:</b> {location}'
if json_res["states"][-1].get("status"):
status = json_res["states"][-1]["status"]
message += f'\n<b>Статус:</b> {status}'
if json_res["states"][-1].get("service"):
serviceName = json_res["states"][-1]["service"]["name"]
message += f'\n<b>Поштовий сервіс: </b>{serviceName}'
return message

```

ДОДАТОК Б
Демонстраційний матеріал

