Лабораторна робота №12

Мета роботи: ознайомитися з роботою з бібліотек для написання ботів для месенджера Telegram.

Завдання 1. Бот починає роботу з команди “/start” і послідовно задає користувачу 10 тестових питань з чотирма варіантами відповіді.

Завдання 2. Питання мають стосуватися мови python, особливості ООП на мові python, її бібліотек, типових конструкцій коду тощо.

Завдання 3. Варіанти відповідей можуть вказуватись як текстово, так і у вигляді кнопок.

Завдання 4. При кожному запуску боту питання і відповіді мають задаватись користувачу у випадковому порядку.

Завдання 5. Правильність чи хибність відповіді користувача на тестове питання має відображатись або відразу після кожного питання, або після повного складання тесту у формі звіту за його результатами.

Завдання 6. Результат складання тесту має містити кількість набраних балів, дату і тривалість проходження тесту.

Лістинг:

import telebot  
from text\_question import questions, patter\_for\_true, patter\_for\_false, answers  
import random  
from telebot import types  
from datetime import datetime  
from time import time  
  
token = '5452367281:AAGoH1z4781CS4OVWqFZow7JDOmLdho\_gX4'  
bot = telebot.TeleBot(token)  
ans\_id = 0  
  
  
@bot.message\_handler(commands=['start'])  
def start(message):  
 start\_time = round(time())  
 global flag, complete, total  
 complete = False  
 total = 0  
 if message.text == "/start":  
 bot.send\_message(message.from\_user.id,  
 "Hello, it is bot for testing.I have 10 questions for you.And we start now!")  
 random.shuffle(questions)  
 for i in range(len(questions)):  
 random.shuffle(questions[i].variants)  
 bot.send\_message(message.chat.id, questions[i].text, reply\_markup = buttons(questions[i].variants))  
 flag = 1  
 while flag == 1:  
 pass  
 if i == len(questions) - 1:  
 complete = True  
 markup = types.ReplyKeyboardRemove(selective = False)  
 test\_time = round(time()) - start\_time  
 bot.send\_message(message.chat.id, f"You pass the test.🎉\nYour time is {test\_time} seconds.⌛\nToday date:{datetime.now().strftime('%d-%m-%Y,%H:%M:%S')}.\nYour goal is {total}/10!👍\nCongrats!🥳", reply\_markup = markup)  
  
  
def buttons(variants):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True, row\_width=2)  
 btn1 = types.KeyboardButton(variants[0])  
 btn2 = types.KeyboardButton(variants[1])  
 btn3 = types.KeyboardButton(variants[2])  
 btn4 = types.KeyboardButton(variants[3])  
 markup.add(btn1, btn2, btn3, btn4)  
 return markup  
  
  
def check(message):  
 global flag,total  
 random.shuffle(patter\_for\_true)  
 random.shuffle(patter\_for\_false)  
 if message.text in answers:  
 bot.send\_message(message.chat.id, patter\_for\_true[0])  
 flag = 0  
 total += 1  
 else:  
 bot.send\_message(message.chat.id, patter\_for\_false[0])  
 flag = 0  
  
  
@bot.message\_handler(func=lambda message: True)  
def echo\_all(message):  
 if not complete:  
 check(message)  
 else:  
 bot.send\_message(message.chat.id,"Your are finish test.Enter the '/start', if you want pass the test again!")  
  
  
bot.polling(none\_stop=True)

Клас:

from question import Question  
  
Question1 = Question("Яка з бібліотек для написання телеграм-ботів використовує asyncio?",  
 ["python-telegram-bot", "Aiogram", "PyTelegramBotApi", "telebot"])  
Question2 = Question("Яка функція використовується для перетворення Jinja2 шаблону в html-сторінку у Flask-додатках?",  
 ["render\_html()", "print\_template()", "app.route()", "render\_template()"])  
Question3 = Question("В якому об’єкті зберігаються значення між запитами окремого користувача до Flask-додатку?",  
 ["request", "route", "current\_app", "session"])  
Question4 = Question(  
 "Яка інструкція в шаблонізаторі Jinja2 використовується для вказування на батьківський шаблон при успадкуванні?",  
 ["extends", "set", "render", "import"])  
Question5 = Question("Який об'єкт відслідковує стани і керує хендлерами в aiogram?",  
 ["Broker", "Mayor", "Dispatcher", "Master"])  
Question6 = Question(  
 "Який шлях (route) за замовчуванням завжди додається до списку шляхів, що містяться в атрибуті url\_map примірнику Flask?",  
 ["'app.py'", "'main.py'", "'/templates/index.html'", "'/static/<filename>'"])  
Question7 = Question(  
 "В якому каталозі за замовчуванням зберігаються CSS, JavaScript та файли шрифтів в Flask-додатках?",  
 ["static", "css", "app", "flask\_app"])  
Question8 = Question("Який шаблонізатор за замовчуванням використовується веб-фреймворком Flask?",  
 ["Genshi", "Jinja2", "Handelbars", "Smarty"])  
Question9 = Question("Якою конструкцією визначаються вирази в шаблонізаторі Jinja2 ?",  
 ["{{ }}", "{% %}", "# #", "{# #}"])  
Question10 = Question(  
 "Стандарт, що визначає API Python для доступу до баз даних різного типу і використовуєтсья у SQLALchemy називається:",  
 ["SQLite", "PyAPI", "SQLAlchemyAPI", "DBAPI"])  
  
questions = [Question1, Question2, Question3, Question4, Question5, Question6, Question7, Question8, Question9,  
 Question10]  
answers = ["Aiogram",  
 "render\_template()",  
 "session",  
 "extends",  
 "Dispatcher",  
 "'/static/<filename>'",  
 "static",  
 "Jinja2",  
 "{{ }}",  
 "DBAPI"]  
patter\_for\_true = ["Right answer\u2714", "Wowww,you good today!", "Good.Continue on the same way!\u2705",  
 "Niceeeeeee,strong knowledge\u2714"]  
patter\_for\_false = ["False", "Try again\u2716", "Not today", "It is not correct"]

Результат виконання:

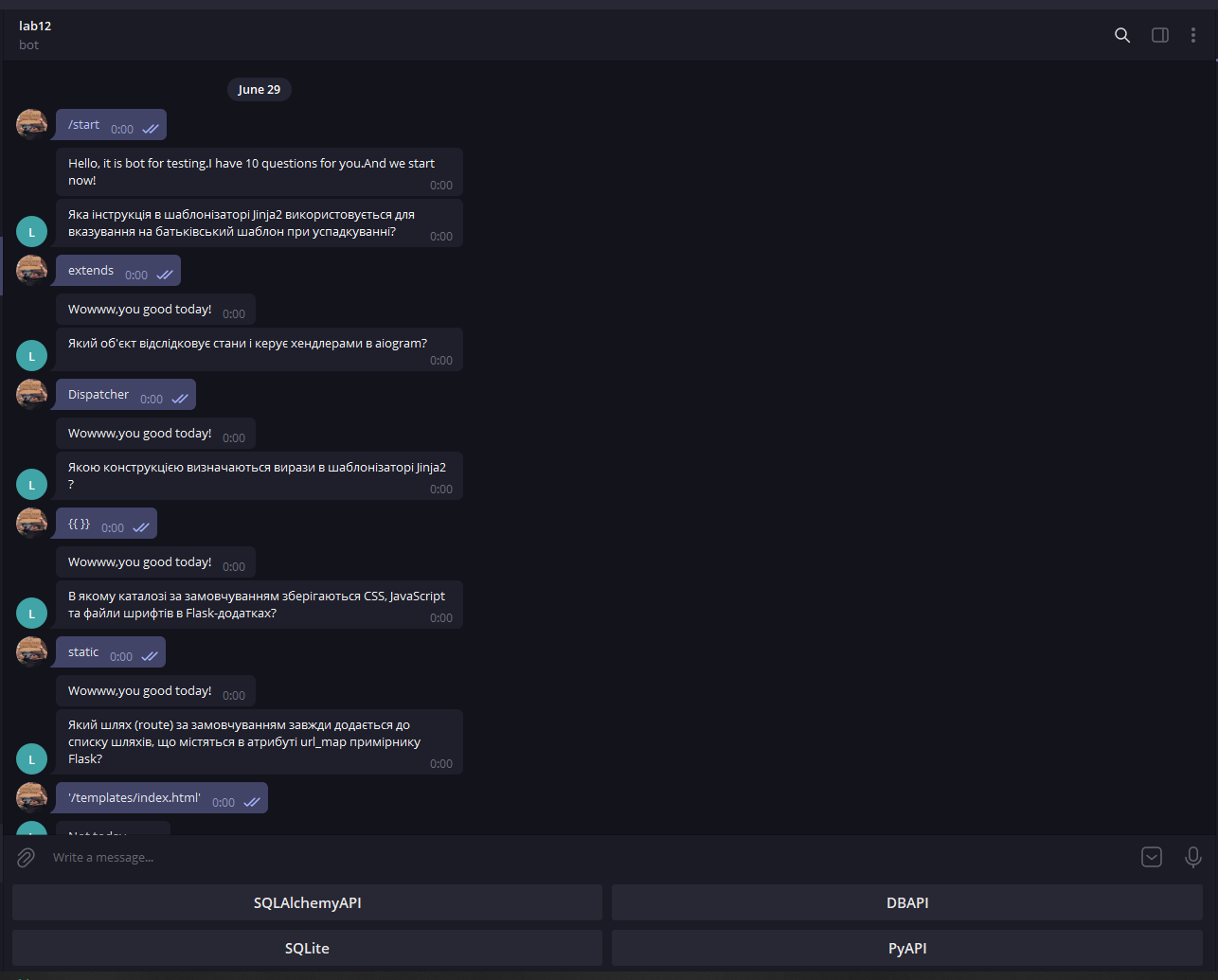


Рис.1.Результат виконання

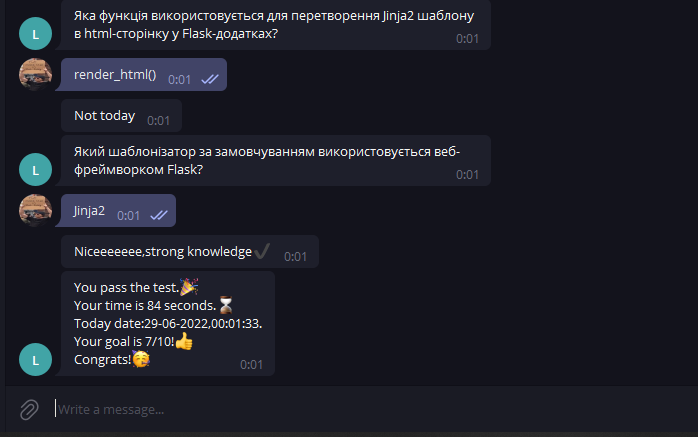


Рис.2.Результат виконання

Висновок:на даній лабораторній роботі,я ознайомився з роботою з бібліотек для написання ботів для месенджера Telegram.