**Лабораторная работа №4. Реализация Telegram-бота на языке программирования Python**

**Цель работы**: получение навыков создания Telegram-ботов на языке программирования Python.

**Примечание к работе:**

Данная лабораторная работа состоит из трех логических блоков: усовершенствование логики работы Telegram-бота, перенос Telegram-бота на webhook и внедрение новой логики в Telegram-бота на webhook. Лабораторную работу возможно выполнять в паре, соответственно один из участников усовершенствует логику работы Telegram-бота, а второй – настраивает окружение и переносить Telegram-бота на webhook.

**Задание на работу (усовершенствование Telegram-бота)**:

1. Выделить в текущей реализации Telegram-бота из л.р. 3 недостатки хранения пользовательских данных. Описать их в отчёте, описать к чему они могут привести, предложить способы противодействовать выделенным недостаткам.
2. Усовершенствовать метод хранения данных в соответствии с выделенными недостатками.
3. Добавить роль администратора в Telegram-бота. Администратор через Telegram-бота способен:

* просмотреть список пользователей и количество выполненных для него предсказаний;
* удалить пользователя;
* добавить нового администратора.

Способ добавления первого администратора, отображение пользователей и выполнение действий остаётся на усмотрение студента (например, первым администратором становиться первый написавший боту или тот, кто введёт сгенерированное значение, отображение просто сообщением и для выбора действий отсылать номер пользователя или сообщение с кнопками для выбора действия).

1. В отчете описать выбранные способы реализации, их преимущества и недостатки. Привести примеры работы, описать ключевые моменты реализации.

**Задание на работу (Перенос на webhook)**:

1. Изучить функционал webhook из Telegram API (<https://core.telegram.org/bots/api#setwebhook>, <https://core.telegram.org/bots/webhooks>). Кратко описать основные положения.
2. При помощи сервиса pythonanywhere.com (либо любого другого сервиса для хостинга веб-приложений) создать веб-приложение, которое будет содержать конечные точки, выступающие в роли webhook’ов. В качестве фреймворка для веб-приложения рекомендуется использовать Flask (<https://flask.palletsprojects.com/en/2.3.x/quickstart/>). Описать выбранный сервис, описать выбранный фреймворк.
3. Веб-приложение должно содержать обработчик POST-запросов. На данный обработчик в дальнейшем будут приходить запросы от Telegram API в формате JSON.
4. Зарегистрировать созданный webhook в Telegram-боте при помощи Telegram API (setWebhook). Описать процесс регистрации.
5. Отправить сообщение в чат бота, использующего webhook. Ознакомиться с форматом полученного веб-приложением сообщения.
6. Реализовать обработку получаемых сообщений и отправку ответных сообщений при помощи API sendMessage (<https://core.telegram.org/bots/api#sendmessage>). Описать формат получаемых сообщений и как происходит отправка ответных сообщений.

**Задание на работу (Внедрение)**:

1. Перенести усовершенствованного Telegram-бота (функционал из л.р. 3 должен сохраниться) на webhook’и. Описать основные моменты переноса.

**Контрольные вопросы:**

1. Какие преимущества и недостатки реализаций Telegram-ботов с использованием polling’a и webhook’ов.
2. Для чего используются хэш-функции при работе с паролями?
3. Что такое «белый» IP-адрес? Зачем применяется?
4. Для чего применяются базы данных?
5. Для чего применяется JSON формат файлов?