Московский государственный технический университет им. Н.Э. Баумана

Факультет ИУ
Кафедра ИУ5

Курс «Основы информатики»
Отчет по Лабораторной работе

Выполнил студент группы ИУ5-33Б: Уфимцев Е.Е. Подпись и дата:

Проверил преподаватель каф.: Гапанюк Ю. Е. Подпись и дата:

Москва, 2024 г

Telegram bot с использованием кнопочного интерфейса и реализацией через библиотеку python-telegram-bot

Code Realisation:

```
from typing import Final
from telegram import Update, ReplyKeyboardMarkup,
ReplyKeyboardRemove, KeyboardButton
from telegram.ext import Application, CommandHandler,
MessageHandler, filters, ContextTypes


TOKEN: Final =
'7623110363:AAGyFDqqTAvTPoHms8wdABEFaWJUQ4xnBEo'
USERNAME: Final = '@bookmarks4sheron_bot'


# Commands
async def start_command(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    keyboard = [
        ['Hello', 'Help'],
        ['GitHub', 'Custom command']
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard,
resize_keyboard=True)

    await update.message.reply_text('Terminal: <Status>
... Vault activated', reply_markup=reply_markup)


async def help_command(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text('Terminal:
<Help> ... waiting for tasks ...')


async def custom_command(update: Update, context:
ContextTypes.DEFAULT_TYPE):
```

```
    await update.message.reply_text('Terminal:
<Command> ...')


# Responses
def handle_response(response: str) -> str:
    text: str = response.lower()
    if 'is anybody here?' in text:
        return '...Sound of silence...'
    elif text == 'help':
        return 'Terminal: <Help> ... waiting for
tasks ...'
    elif text == 'custom command':
        return 'Terminal: <Command> ...'
    elif text == 'github':
        return 'Github repo: https://github.com/Sheron-
Fate/Parsing_'
    elif text == 'hello':
        return 'Terminal: <Status> ...'
    return 'Terminal: <Status> pending messages'


async def handle_message(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    message_type = update.message.chat.type
    text: str = update.message.text

    print(f'User ({update.message.chat.id}) in
{message_type}: "{text}"')

    response: str = handle_response(text)

    print('<Bot>', response)
    await update.message.reply_text(response)


#Errors
async def error(update: Update, context:
ContextTypes.DEFAULT_TYPE):
```

```
    print(f'Update {update} caused error
{context.error}')


def main():
    print('Starting bot')
    app = Application.builder().token(TOKEN).build()

    app.add_handler(CommandHandler('start',
start_command))
    app.add_handler(CommandHandler('help',
help_command))
    app.add_handler(CommandHandler('custom',
custom_command))

    app.add_handler(MessageHandler(filters.TEXT,
handle_message))

    app.add_error_handler(error)

    print('Polling...')
    app.run_polling(poll_interval=3)


if __name__ == "__main__":
    main()
```

Result:


Vault 101
bot
Hello 14:23
Terminal: <Status> ... 14:23
/start 15:33
Terminal: <Status> ... Vault activated 15:33
Hello 15:33
Terminal: <Status> ... 15:33
Help 15:33
Terminal: <Help> ... waiting for tasks ... 15:33
GitHub 15:33
Github repo: https://github.com/Sheron-Fate/Parsing_ 15:33
GitHub 15:34
Github repo: https://github.com/Sheron-Fate/Parsing_ 15:34
Write a message...
Hello
Help
GitHub
Custom command