st { Telegraf } = require('telegraf')

const axios = require('axios')

const TOKEN = '6771352554:AAHovS9YULxVXuT9Ohed9BucKLAIRWxoCrQ'

const SPOONACULAR\_API\_KEY = '6366b43f93c84e438c4a467a35e6c9bc'

const SPOONACULAR\_API\_URL = 'https://api.spoonacular.com'

const bot = new Telegraf(TOKEN)

bot.start((ctx) => ctx.reply('Welcome to Recipe Bot! Send /random to get a random recipe or /recipes to see a list of available recipes.'))

*// bot.help((ctx) => ctx.reply('Send me a sticker'))*

*// Command for authentication*

bot.command('auth', (ctx) => {

const authCode = ctx.message.text.split('/auth ')[1];

if (true) {

ctx.reply(`Authentication successful! Entered code: ${authCode}`);

} else {

ctx.reply('Invalid authorization code. Please try again.');

}

})

*// Command for registration*

bot.command('register', (ctx) => {

const userName = ctx.message.text.split('/register ')[1];

if (userName) {

ctx.reply(`User ${userName} registered successfully!`);

} else {

ctx.reply('Invalid registration format. Please use /register <username>');

}

})

*// Команда для обработки неизвестных команд*

bot.command('unknown', (ctx) => {

ctx.reply('Unknown command. Please use /help for a list of available commands.');

})

*// Command for help*

bot.command('help', (ctx) => {

const helpMessage = `Available commands:

/random - Get a random recipe

/recipes - See a list of available recipes

/cuisine <cuisine\_type> - Get a recipe by cuisine type

/ingredient <ingredient> - Get a recipe by ingredient

/auth <authorization\_code> - Authenticate with the bot

/register <username> - Register with the bot

/unknown - Handle unknown commands

/suggest - Suggest a new feature

/support - Get support

/about - Get information about the bot

/start - Start the bot

/help - Display this help message`

ctx.reply(helpMessage);

})

*// Команда о предложении новой функции*

bot.command('suggest', (ctx) => {

ctx.reply('Thank you for your suggestion! We will review it and get back to you soon.');

})

*// Команда для получения поддержки*

bot.command('support', (ctx) => {

ctx.reply('Please contact us at support@recipebot.com for any support issues.');

})

*// Команда для получения информации о боте*

bot.command('about', (ctx) => {

ctx.reply('Recipe Bot is a bot that helps you find recipes based on cuisine type or ingredient.');

})

*// Функция получения случайного рецепта*

bot.command('random', *async* (ctx) => {

try {

const response = await axios.get(`${SPOONACULAR\_API\_URL}/recipes/random?apiKey=${SPOONACULAR\_API\_KEY}&number=1`)

const recipe = response.data.recipes[0]

const recipeMessage = `

🍽 \*${recipe.title}\*

📷 [${recipe.sourceName}](${recipe.sourceUrl})

📝 ${recipe.instructions.split('\n').map(line => `- ${line}`).join('\n')}

📷 [${recipe.image}](${recipe.image})

`

ctx.replyWithMarkdown(recipeMessage)

} catch (error) {

console.error(error)

ctx.reply('Sorry, something went wrong. Please try again later.')

}

})

*// Функция для получения списка доступных рецептов*

bot.command('recipes', (ctx) => {

const recipes = [

'Italian',

'Mexican',

'Chinese',

'Indian',

'Thai',

'Japanese',

'Vegan',

'Vegetarian',

'Gluten-free',

'Desserts',

]

const recipeList = recipes.join(', ')

ctx.reply(`Available recipes: ${recipeList}`)

})

*// Функция получения рецепта по типу кухни*

const cuisineTypes = {

Italian: '351',

Mexican: '365',

Chinese: '5',

Indian: '23',

Thai: '21',

Japanese: '11',

Vegan: '701',

Vegetarian: '703',

Gluten\_free: '306',

Desserts: '12',

}

bot.command('cuisine', *async* (ctx) => {

const cuisine = ctx.message.text.split('/cuisine ')[1].toLowerCase()

if (!cuisineTypes[cuisine]) {

ctx.reply('Sorry, that cuisine type is not available. Please try again.')

return

}

try {

const response = await axios.get(`${SPOONACULAR\_API\_URL}/recipes/complexSearch?apiKey=${SPOONACULAR\_API\_KEY}&cuisine=${cuisineTypes[cuisine]}&number=1`)

const recipe = response.data.results[0]

const recipeMessage = `

🍽 \*${recipe.title}\*

📷 [${recipe.sourceName}](${recipe.sourceUrl})

📝 ${recipe.instructions.split('\n').map(line => `- ${line}`).join('\n')}

📷 [${recipe.image}](${recipe.image})

`

ctx.replyWithMarkdown(recipeMessage)

} catch (error) {

console.error(error)

ctx.reply('Sorry, something went wrong. Please try again later.')

}

})

*// Функция получения рецепта по ингредиентам*

bot.command('ingredient', *async* (ctx) => {

const ingredient = ctx.message.text.split('/ingredient ')[1].toLowerCase()

try {

const response = await axios.get(`${SPOONACULAR\_API\_URL}/recipes/complexSearch?apiKey=${SPOONACULAR\_API\_KEY}&ingredients=${ingredient}&number=1`)

const recipe = response.data.results[0]

const recipeMessage = `

🍽 \*${recipe.title}\*

📷 [${recipe.sourceName}](${recipe.sourceUrl})

📝 ${recipe.instructions.split('\n').map(line => `- ${line}`).join('\n')}

📷 [${recipe.image}](${recipe.image})

`

ctx.replyWithMarkdown(recipeMessage)

} catch (error) {

console.error(error)

ctx.reply('Sorry, something went wrong. Please try again later.')

}

})

bot.launch()