Министерство образования и науки Челябинской области

Государственное бюджетное профессиональное образовательное учреждение

**«Южно-Уральский государственный технический колледж»**

|  |  |
| --- | --- |
| ЗАЩИЩЕНО | |
| ОЦЕНКА: | |
|  | Ильина О.С. |
|  | Женихова И.Ю. |
|  | Пигаль К.А. |
|  | Шибанова В.А. |

**ОТЧЕТ**

о прохождении учебной практики УП.09

по ПМ.09 Проектирование, разработка и оптимизация веб-приложений

специальность

 09.02.07 Информационные системы и программирование

(квалификация: разработчик веб и мультимедийных приложений**)**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Разработал: | студент группы ВБ-464 |
|  | Сметанин А.Ю. |

Челябинск, 2022

Содержание

Раздел 1. Разработка веб-приложения (Ильина О.С. и Женихова И.Ю.)

Разработка клиентской части (?исправить)

Разработка телеграм-бота

Раздел 2. SEO оптимизация сайта (Пигаль К.А.)

Раздел 3. CMS WordPress (Шибанова В.А.)

Раздел 1. Разработка веб-приложения

Клиентская часть (?)

Разработка телеграм-бота

Во время прохождения учебной практики был разработан телеграм-бот, позволяющий получать новостные рассылки с различных ресурсов.

Для разработки использовались sqlite3, pyTelegramBotAPI, newsAPI, requests.

Функционал бота включает возможность регистрации, подписки, отписки, просмотра новостей по выбранным категориям.

Структура:

* bot.py. Основной файл, в котором происходит взаимодействие с пользователем.
* create\_db.py. В нем хранится функция создания базы данных.
* functions\_db.py. Файл, в котором содержатся функции для взаимодействия с базой данных и пользователем.
* markups.py. Файл, в котором содержится функция для создания кнопок.
* config.py. В нем хранятся токены.
* requirements.txt. Файл зависимостей приложения.

Ссылка на репозиторий с ботом: https://github.com/psixoz001/TelegramBot

Код:

import sqlite3

import telebot

from config import \*

from telebot import types

from create\_db import \*

from markups import \*

from functions\_db import \*

import requests

bot = telebot.TeleBot(bot\_token)

@bot.message\_handler(commands=['start'])

def start(message):

user = findUser(message.from\_user.id)

if user is None:

registration(message.from\_user.id)

connect.commit()

bot.send\_message(message.chat.id, f'Здравствуйте, {message.from\_user.username}', reply\_markup=anyButtons(["Мои подписки","Подписаться","Отписаться","Новости"]))

@bot.message\_handler(content\_types=['text'])

def func(message):

if(message.text == "Мои подписки"):

mySubs = findSubs(message.from\_user.id)

if(len(mySubs)>0):

mySubsText=' '.join(''.join(tup) for tup in mySubs)

bot.send\_message(message.chat.id,' '.join(''.join(tup) for tup in mySubs))

else:

bot.send\_message(message.chat.id, 'У вас нет подписок')

elif(message.text == "Подписаться"):

btns=[]

for key in category:

btns.append(f'sub\_{key}')

bot.send\_message(message.chat.id,'Выберите категорию, на которую хотите подписаться: ', reply\_markup=anyButtons(btns))

elif(message.text == "Отписаться"):

mySubs = findSubs(message.from\_user.id)

if(len(mySubs)>0):

btns=[]

for item in mySubs:

btns.append(f'unsub\_{item[0]}')

bot.send\_message(message.chat.id,'Выберите категорию, от которой хотите отписаться: ', reply\_markup=anyButtons(btns))

else:

bot.send\_message(message.chat.id, 'У вас нет подписок')

elif(message.text in ['sub\_business', 'sub\_entertainment','sub\_general','sub\_health','sub\_science','sub\_sport','sub\_technology']):

category\_name = message.text.split('\_')[1]

category\_id=gimmeIdCategory(str(category\_name))[0]

if(findSubsAll(message.from\_user.id,category\_id)) is None:

subscribe(message.from\_user.id, category\_id)

bot.send\_message(message.chat.id,f'Вы подписались на категорию {category\_name}',reply\_markup=anyButtons(["Мои подписки","Подписаться","Отписаться","Новости"]))

else:

bot.send\_message(message.chat.id,'Вы уже подпипсаны на эту категорию',reply\_markup=anyButtons(["Мои подписки","Подписаться","Отписаться","Новости"]))

elif(message.text in ['unsub\_business', 'unsub\_entertainment','unsub\_general','unsub\_health','unsub\_science','unsub\_sport','unsub\_technology']):

category\_name = message.text.split('\_')[1]

category\_id=gimmeIdCategory(str(category\_name))[0]

unsubscribe(message.from\_user.id, category\_id)

bot.send\_message(message.chat.id,f'Вы отписались от категории {category\_name}',reply\_markup=anyButtons(["Мои подписки","Подписаться","Отписаться","Новости"]))

elif(message.text == "Новости"):

mySubs = findSubs(message.from\_user.id)

if(len(mySubs)>0):

btns=[]

for item in mySubs:

btns.append(f'{item[0]}')

bot.send\_message(message.chat.id,'Выберите категорию, по которой хотите увидеть новости: ', reply\_markup=anyButtons(btns))

elif(message.text in ['business', 'entertainment','general','health','science','sport','technology']):

response=requests.get(f'https://newsapi.org/v2/top-headlines?apiKey={news\_token}&category={message.text}&pageSize=10&country=ru')

for i in response.json()['articles']:

bot.send\_message(message.chat.id,f"{i['title']},\n{i['publishedAt']},\n{i['url']}")

bot.send\_message(message.chat.id,'Выберите следующее действие',reply\_markup=anyButtons(["Мои подписки","Подписаться","Отписаться","Новости"]))

createDB()

bot.polling()

Раздел 2. SEO-оптимизация

Раздел 3. CMS WordPress