

Разработка оптимизации поиска ближайших
сортировочных пунктов отходов по России благодаря
возможностям мессенджера Telegram
и информационный сайт о решении экологических
проблем в мировом масштабе

Сурнина Анастасия, 10 класс ГБОУ Цифровая школа
Иванов Антон, 10 класс ГБОУ Цифровая школа
Руководитель: Донцова Мария Александровна, учитель



Актуальность

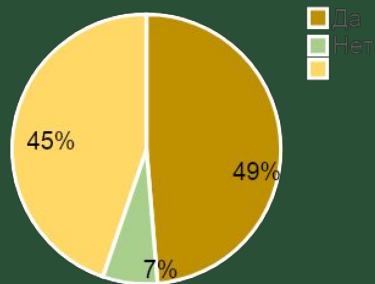
Сегодня экологические проблемы касаются практически каждого человека. В российских городах уровень загрязнения в десятки раз превышает норму, поэтому необходимо использовать все методы борьбы с данной проблемой и одним из таких методов является сортировка отходов.

Сортировка облегчает переработку мусора, а также позволяет частично отказаться от мусоросжигания.



Результаты анкетирования

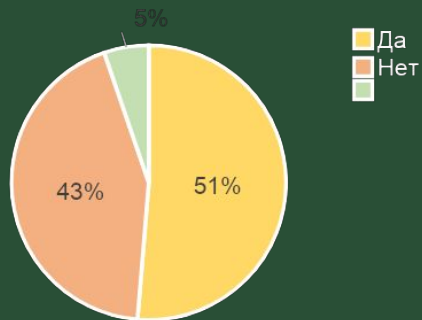
Знаете ли вы о проблемах с экологической обстановкой в вашем регионе?



Вы знаете, где найти ближайший пункт раздельного сбора мусора?

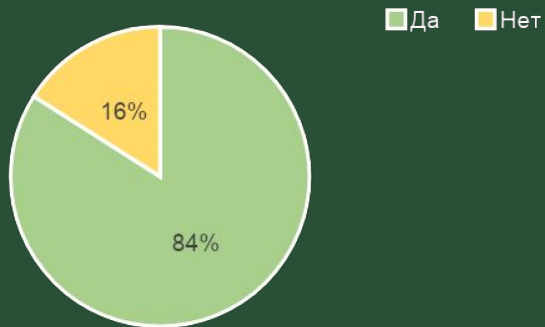


Пользуетесь ли вы пунктами сортировки мусора?

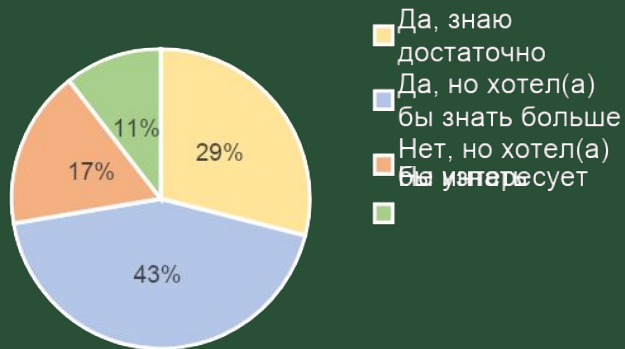


Результаты анкетирования

Вы бы стали правильно сортировать мусор, если бы информация о местоположении пунктов сортировки была бы более доступной?



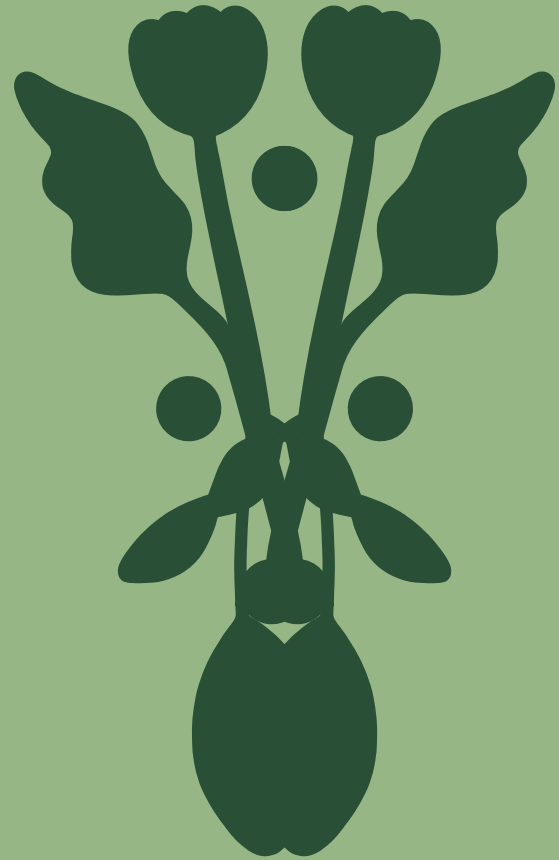
Вы знаете, как решаются экологические проблемы в других странах?



Цель работы:

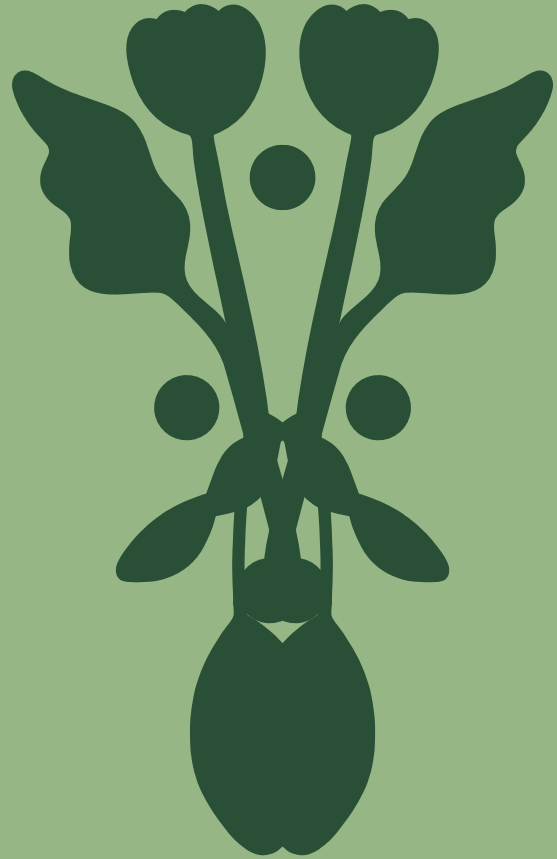


Создать новые информационные ресурсы в сети интернет для помощи населению России в решении проблемы раздельного сбора и вторичной переработки отходов



Задачи работы:

- Исследовать состояние проблемы проведя социологический опрос с применением вероятностно-статистического метода исследований.
- Создать информационный сайт о проблемах раздельного сбора и вторичной переработки отходов.
- Разработать чат-бот, который поможет жителям России найти ближайшие пункты приема отходов, подлежащих вторичной переработке.
- Распространить информацию о созданных продуктах в социальных сетях для привлечения пользователей.
- Определить перспективы развития проекта.





Анализ существующих решений

<https://recyclemap.ru/>

На этом сайте есть карта сортировочных пунктов, но неудобно указывать местоположение пользователя



GREENPEACE ПОМОЩЬ СДАТЬ ЕМКУ НА ПЕРЕРАБОТКУ СПАСТИ РАЗДЕЛЬНЫЙ СБОР И ПЕРЕРАБОТКУ НАПИСАТЬ МОДЕРАТОРУ УЗНАТЬ БОЛЬШЕ ВХОД

Москва

Где вы находитесь?

ВЫБЕРИ, ЧТО ХОЧЕШЬ СДАТЬ:

- Бумага
- Пластик
- Стекло
- Металл
- Текстиль
- Смешанный мусор
- Смешанный пластик
- Смешанный металл
- Смешанный текстиль
- Смешанный стекло
- Смешанный бумага
- Смешанный пластик
- Смешанный металл
- Смешанный текстиль
- Смешанный стекло
- Смешанный бумага

Показать пункты, которые принимают сразу все выбранное

БЛИЖАЙШЕ

КОНТЕЙНЕР ДЛЯ ПРИЕМА БАНОЧЕК ИЗ-ПОД ПРОДУКЦИИ ЗВАЛАР В АПТЕКЕ "ЗВАЛАР"
Москва, Таврическая ул., 4

© 2018 Greenpeace Russia. Карта: Mapbox © OpenStreetMap

Анализ возможностей социальных сетей



Мгновенная скорость работы

Более 200
доступных
функций

Визуал ничем не
ограничен

Обновления
каждые 3 дня

Функционал
ограничен
условностями
программы

Максимальная
защита данных



Более 150
доступных функций

Визуал ограничен дизайном программы

Обновления
каждую неделю

Функционал
ограничен
особенностями
соглашения

Защита данных с поправкой на соглашение



Имеются задержки

Более 50
доступных функций

Обновления
каждый месяц

Функционал ничем
не ограничен

Материалы

Программа PyCharm



Язык программирования Python



Язык программирования JavaScript



Язык программирования HTML



Сервис Yandex Cloud



Yandex Cloud

Методы

- **Проведение социологического опроса с применением вероятностно-статистического метода исследований.** Был создан опросный лист в Google Forms. Описание респондентов: возраст от 14 до 40 лет, общее количество – 180 человек
- **Создание сайта «Как в разных странах сортируют мусор?».** Написание текста и создание шаблона сайта на HTML. Анимация элементов на JavaScript
- **Выбор социальной сети, для которой будет написан чат-бот.** Характеристики были выбраны исходя из комфортности для пользователя и обработке запросов для ускоренной работы чат-бота
- **Выбор языка программирования и создание чат-бота для Telegram.** Для написания кода для чат-бота использован язык программирования Python
- **Размещение кода для чат-бота на сервисе Yandex Cloud**
- **Выполнение тестирования и отладки сайта и чат-бота**
- **Распространение информации о созданных продуктах в социальных сетях для**

Перспективы развития



Продвижение

Продвижение в массы мою продукцию (практически повсеместное использование по всей России)



Функционал

Планируется увеличить функционал чат-бота (выбор конкретного вида отходов)



Авто определение отходов

Ввести возможность распознавать тип отходов с помощью штрих-кодов на упаковке, чтобы указывать пользователю пункты сортировки только для данного типа



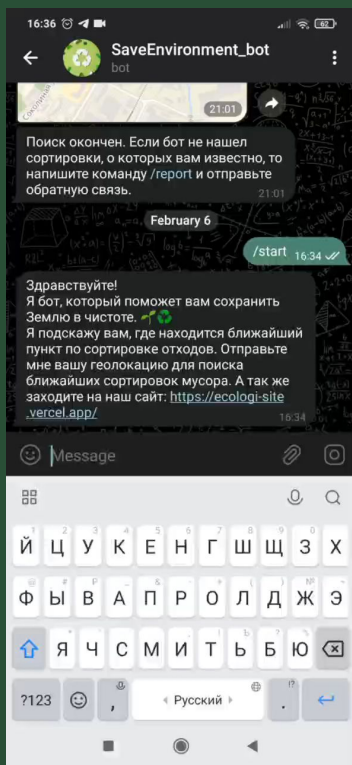
Выводы

- **Социологический опрос** показал, что больше половины респондентов заинтересованы в получении дополнительной информации о мировых решениях экологических проблем и ближайших пунктов приема вторичного сырья.
- Нами создан сайт «**Как в разных странах сортируют отходы?**», который позволяет получить дополнительную информацию о сортировке и переработки вторсырья в разных странах.
- Нами разработан чат-бот в мессенджер **Telegram**, который помогает найти ближайшие пункты приема вторсырья.
- Для распространения информации о наших продуктах мы использовали социальную сеть **ВКонтакте**, а также свой канал в мессенджере **Telegram**.
- **Перспективы развития проекта** связаны с продвижением сайта и чат-бота для широкого использования и увеличением функционала чат-бота.

Итог

Созданные ресурсы помогают эффективно бороться с **одной из проблем – с неправильной сортировкой отходов**. Создав эту программу нам бы хотелось **помочь городу поддерживать чистоту** через популярные и доступные способы получения информации для граждан. Тем самым наш продукт полностью направлен на **сокращение влияния** данной проблемы на большие города, обеспечив их свободным ресурсом, **доступным практически каждому гражданину** моей страны





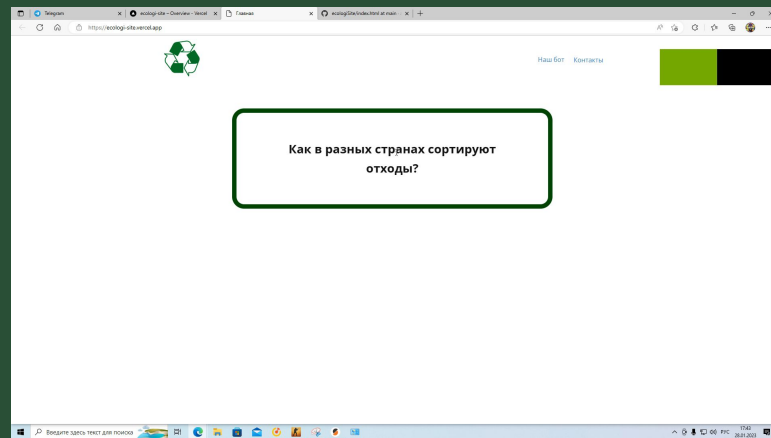
Чат-бот:



Сайт:



Результаты:



Спасибо за
уделенное время

Surnina.Anastassiya@ya.ru
+7(968)664-5501

antonianton2000@ya.ru
+7(977)727-5739



Программный код. Telegram-бот

```
import telebot
import json
import requests
import json

bot = telebot.TeleBot('5318941676:AAE6SA0Z11y1YJJmaJr1PoJ2yM41xMpTVLo')

flag = True

@bot.message_handler(content_types=['text'])
def start(message):
    global flag
    if message.text == '/start':
        bot.send_message(message.from_user.id, '''Здравствуйте!  
Я бот, который поможет вам сохранить Землю в чистоте. ♻️  
Я подскажу вам, где находится ближайший пункт по сортировке отходов. Напишите /find для нахождения ближайших сортировок. А Т
    elif message.text == '/find':
        bot.send_message(message.from_user.id, "Отправьте мне вашу геолокацию")
    elif message.text == '/report':
        bot.send_message(message.from_user.id, "Отправьте мне геолокацию сортировки.")
        flag = False
    elif message.text == 'aaaddd':
        with open('reports.json', 'r', encoding='utf-8') as fh:
            bot.send_document(message.from_user.id, fh)
```


Программный код. Telegram-бот

```
@bot.message_handler(content_types=['location'])
def find(message):
    global flag, dict_of_reports
    if flag:
        print(message.location.longitude, message.location.latitude)
        bot.send_message(message.from_user.id, 'Начинаю поиск')
        lon, lat = message.location.longitude, message.location.latitude

        dict_of_cit = requests.get('https://xn--1-otblt.xn--plai/map/cm.php?city=-1').json()
        cur_cit = {'id': '1', 'lat': '55.7580321449', 'lng': '37.6168444863', 'title': 'Москва'}
        cur_c = 1000000000
        for i in dict_of_cit:
            if (abs(lat - float(i['lat'])) ** 2 + abs(lon - float(i['lng'])) ** 2) ** 0.5 < cur_c:
                cur_cit = i
                cur_c = (abs(lat - float(i['lat'])) ** 2 + abs(lon - float(i['lng'])) ** 2) ** 0.5
        id = cur_cit['id']

        dict_of_rec = requests.get(f'https://сmw1.p0/map/cm.php?city={id}').json()
        result = []
        metr = 0.0001
        while len(result) <= 5:
            result = []
            metr += 0.0001
            for i in dict_of_rec:
                lat1, lon1 = float(dict_of_rec[i]['lat']), float(dict_of_rec[i]['lng'])
                if abs(lon - lon1) <= metr and abs(lat - lat1) <= metr:
                    result.append(dict_of_rec[i])

        print(result)
```

Программный код. Telegram-бот

```
for i in result:
    bot.send_message(message.from_user.id, i['title'])
    bot.send_location(message.from_user.id, *tuple((float(i['lat']), float(i['lng']))))
bot.send_message(message.from_user.id,
                 'Поиск окончен. Если бот не нашел сортировки, о которых вам известно, то напишите команду')
else:
    with open('reports.json', 'r', encoding='utf-8') as fh:
        dict_of_reports = json.load(fh)
    dict_of_reports[str(len(dict_of_reports))] = str(message)
    with open('reports.json', 'w', encoding='utf-8') as fh:
        fh.write(json.dumps(dict_of_reports, ensure_ascii=False))
    flag = True
    bot.send_message(message.from_user.id,
                    'Спасибо за обратную связь')

bot.polling(none_stop=True)
```

Программный код. Сайт

```
from flask import Flask, render_template, request, url_for, redirect, jsonify, request

app = Flask(__name__)

@app.route('/')
def hello():
    return redirect(f'https://blooming-headland-96587.herokuapp.com/static/index.html')

if __name__ == '__main__':
    app.run(debug=True)
```