# ВАДИМ КРУПИНИН

РАЗРАБОТЧИК НА **PYTHON**

Email: vadimkrupinin@gmail.com

Telegram: @KVadim_K

Github: https://github.com/KVadim-K

# О СЕБЕ

- Окончил ДГТУ
- Окончил курсы по программированию:
  - на **Python** от Зерокодер
  - Интенсив по программированию на **PYTHON** с помощью **chatGPT**
  - Интенсива по разработке мобильных приложений
  - **7**-дневная групповая разработка на **Python**
- Имею большой интерес к машинному обучению и разработке веб-приложений.
- Люблю решать сложные и интересные задачи, постоянно развиваюсь в новых технологиях особенно работой с искусственным интеллектом.
- Я вырос в Ростове-на-Дону, увлекаюсь боевыми единоборствами и кроссфитом.
- Ищу интересные предложения!

# НАВЫКИ И ЗНАНИЯ

Базы данных:

MySQL,
Postrgre
SQL,
SQLite

Основной язык программирования:

Pyton

Инструменты и технологии:

Git,
Linux,
Repl.it,
PyCharm

Фреймворки и Библиотеки:

Django
Flask,
Requests,
NumPy,
Pandas

Прочие навыки:

Веб-разработка,
разработка
телеграм-ботов

# ПРОЕКТЫ:

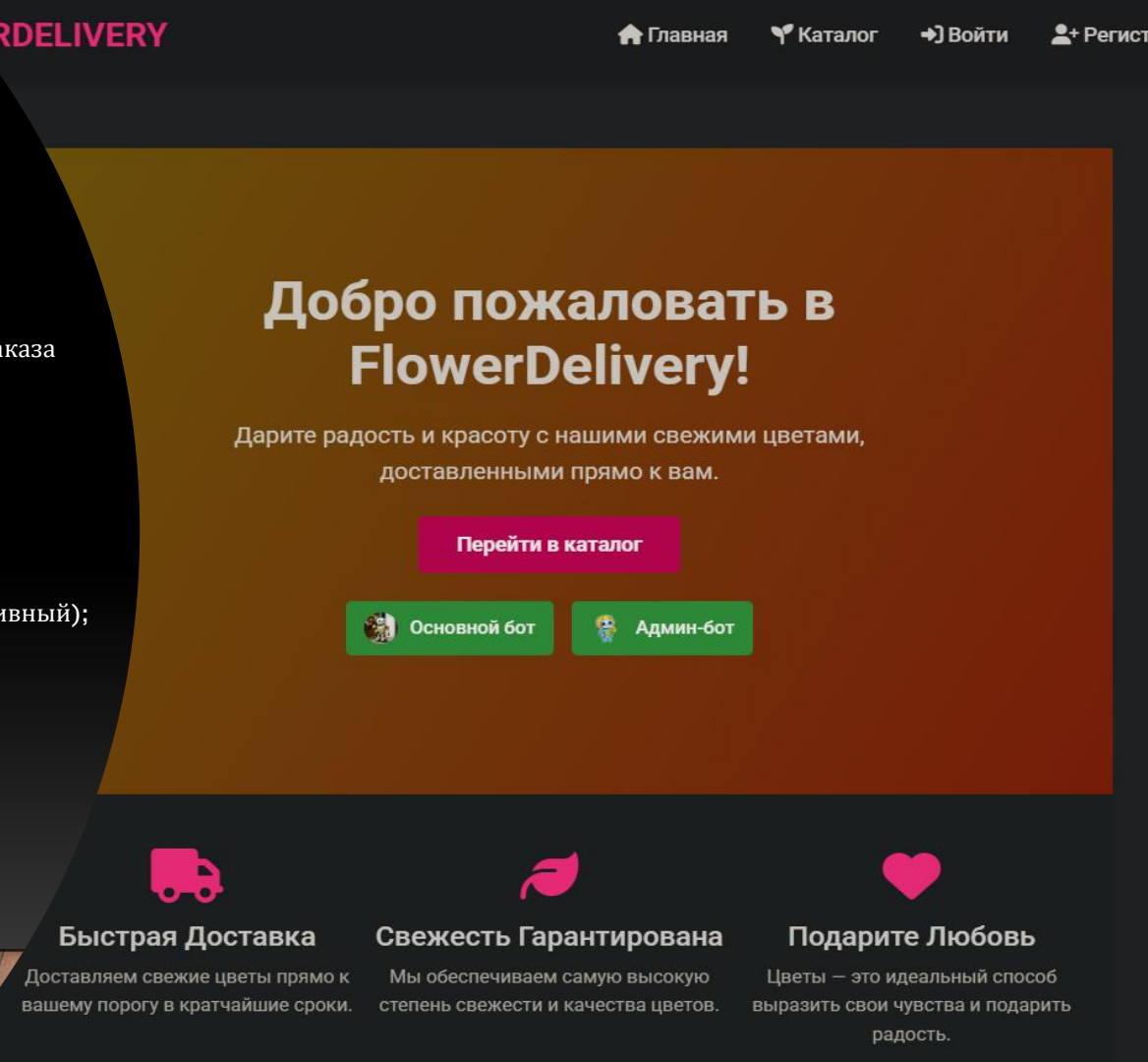
# FLOWER DELIVERY: ОБЩЕЕ ОПИСАНИЕ И ТЕХНОЛОГИИ

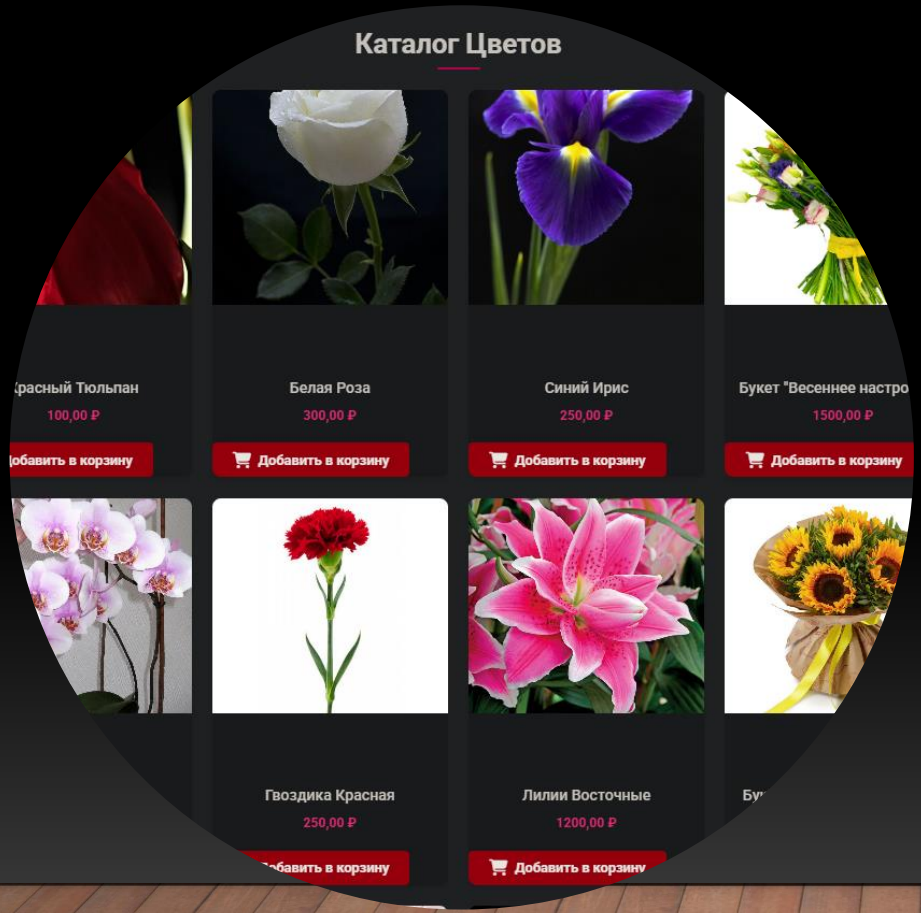
- Flower Delivery — это инновационный сервис для заказа
- и доставки цветов, разработанный
- с использованием современных технологий.

- Проект включает:

- - Telegram-ботов (пользовательский и административный);
- - Систему управления заказами;
- - Инструменты для аналитики и отчетности;
- - Удобный интерфейс для взаимодействия.

- Используемые технологии:
- Django, Redis, Celery, Docker, SQLite, WhiteNoise,
- Telegram API.



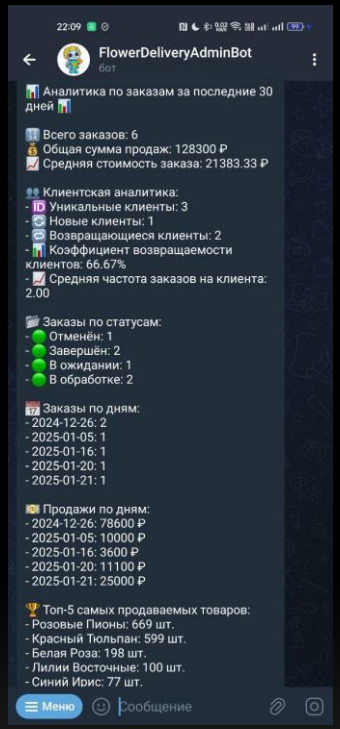


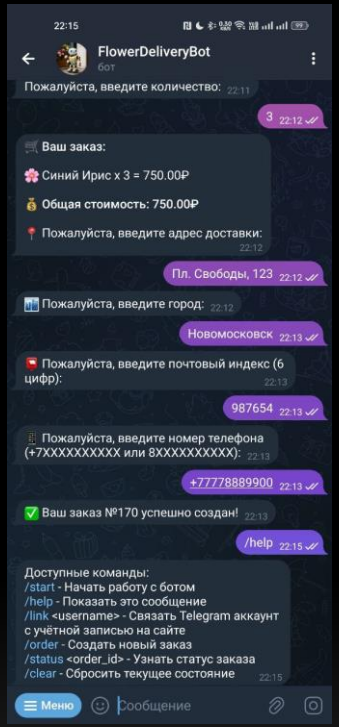

# ЦЕЛЬ, ЗАДАЧИ И ДОСТИЖЕНИЯ

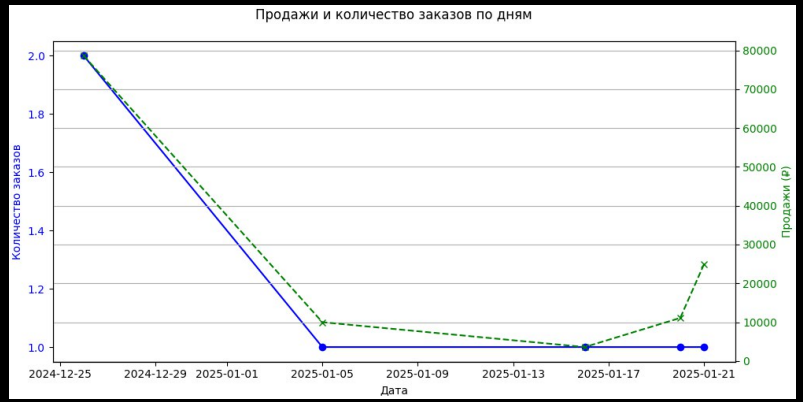
- Цель проекта:
- Создать удобный и масштабируемый сервис для
- автоматизации процесса покупки и доставки цветов.

- Основные задачи:
- - Интеграция Telegram-ботов для пользователей и администраторов;
- - Разработка системы отчетности и аналитики;
- - Настройка фоновых задач через Celery.

- Достижения:
- - Оптимизация процессов доставки;
- - Удобство взаимодействия для конечных пользователей;
- - Высокая производительность и готовность к масштабированию.




# РЕЗУЛЬТАТЫ И ПРИМЕРЫ

- Результаты:
- - Успешно запущенный проект, готовый к масштабированию;
- - Автоматизация задач с помощью **Redis** и **Celery**;
- - Интеграция с **Docker** для контейнеризации.

- Примеры:
- - Диаграмма аналитики заказов на сайте;
- - Скриншоты из интерфейса пользовательского и административного ботов.

# ОЗНАКОМЬТЕСЬ С ПРОЕКТОМ FLOWER DELIVERY

Посетите сайт Flower Delivery и ознакомьтесь с проектом в действии.

Ссылка на сайт: https://vaktest.ru

Или отсканируйте QR-код ниже для быстрого перехода.

# $H_2O$ REMINDER BOT

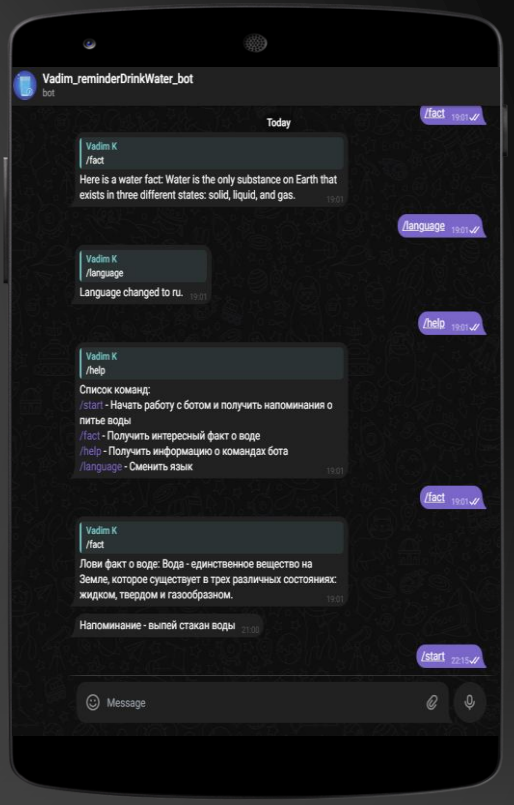
**Задачи проекта:**

***Разработать Telegram-бот,*** который будет напоминать пользователю о необходимости выпить стакан воды каждые 3 часа.

**Описание проекта:**

***H2O Reminder Bot —*** это ваш персональный ассистент по поддержанию водно-солевого баланса. В настоящее время легко забыть о простых, но важных аспектах заботы о себе. Но мой **Telegram**-бот решает эту проблему с помощью регулярных напоминаний о необходимости пить воду.

```python
# Обработчик команды start
@bot.message_handler(commands=['start'])
def start_message(message):
    user_language[message.chat.id] = 'ru'  # Задаем стандартный язык
    bot.reply_to(message, texts[user_language[message.chat.id]]['welcome'])
    reminder_thread = threading.Thread(target=send_reminders, args=(message.chat.id,))
    reminder_thread.start()

# Обработчик команды help
@bot.message_handler(commands=['help'])
def help_message(message):
    bot.reply_to(message, texts[user_language[message.chat.id]]['help'])

# Обработчик команды fact
@bot.message_handler(commands=['fact'])
def fact_message(message):
    language = user_language[message.chat.id]
    random_fact = random.choice(texts[language]['facts'])
    bot.reply_to(message, text: f'{texts[language]["fact"]} {random_fact}')

# Обработчик команды language
@bot.message_handler(commands=['language'])
def change_language(message):
    if user_language[message.chat.id] == 'ru':
        user_language[message.chat.id] = 'en'
    else:
        user_language[message.chat.id] = 'ru'
    bot.reply_to(message, text: f'Language changed to {user_language[message.chat.id]}.')

# Функция отправки напоминаний
1 usage
def send_reminders(chat_id):
    reminders = ["09:00", "12:00", "15:00", "18:00", "21:00"]
    while True:
        now = datetime.datetime.now().strftime('%H:%M')
        if now in reminders:
            bot.send_message(chat_id, texts[user_language[chat_id]]['reminder'])
            time.sleep(60)
        time.sleep(1)

bot.polling(none_stop=True)
```



# TASK TRACKER

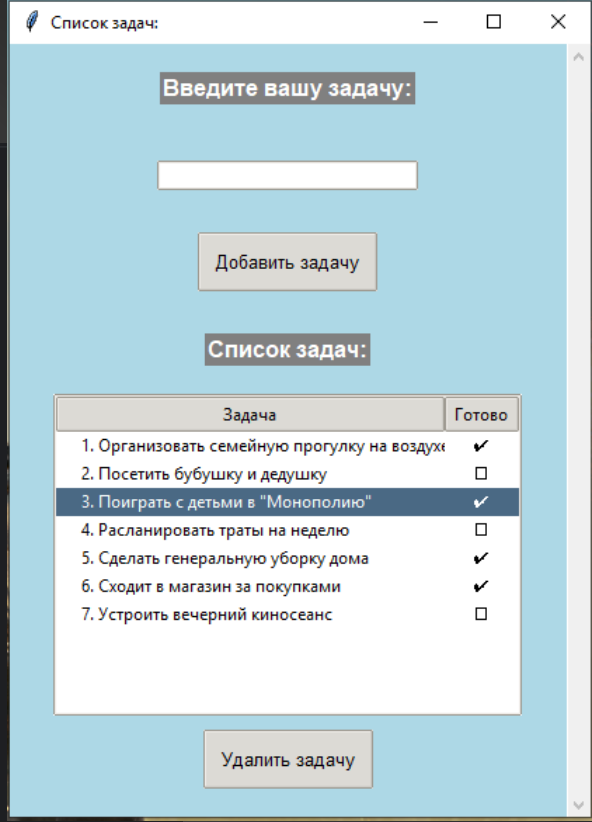
**Задачи проекта:**

***Разработать приложения для управления задачами,*** которое удобно использовать для организации и упорядочивания своих планов на ближайшее время.

**Возможности программы:**

- **добавлять задачи;**
- **помечать задачи выполненными;**
- **удалять задачи.**

```python
import tkinter as tk
from tkinter import ttk

1 usage
def add_task():
    task = task_entry.get()
    if task:
        task_id = task_tree.insert( parent: "", tk.END, text=task, values=("□",))
        task_entry.delete( first: 0, tk.END)

1 usage
def check_task(event):
    item = task_tree.selection()[0]
    checked = task_tree.item(item, option: "values")[0]
    if checked == "□":
        task_tree.item(item, values=("✓",))
    else:
        task_tree.item(item, values=("□",))

1 usage
def delete_task():
    selected_item = task_tree.selection()
    for item in selected_item:
        task_tree.delete(item)

root = tk.Tk()
root.title("Список задач:")
root.geometry("400x550+800+300")
root.configure(bg="lightblue")

style = ttk.Style()
style.theme_use("clam")
style.configure( style: "TLabel", background="gray", foreground="white", font=("Arial", 12))
style.configure( style: "TButton", padding=10, font=("Arial", 10))

scrollbar = tk.Scrollbar(root)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y, expand=False) # здесь мы ставим полоску прокрутки expand = False по умолчанию

text1 = ttk.Label(root, text="Введите вашу задачу:", style="TLabel", font="none 12 bold")
text1.pack(pady=20, padx=5)

task_entry = ttk.Entry(root, width=30)
```



# Я ВСЕГДА ОТКРЫТ К НОВЫМ ПРОЕКТАМ И ВОЗМОЖНОСТЯМ! СВЯЗАТЬСЯ СО МНОЙ МОЖНО СЛЕДУЮЩИМ ОБРАЗОМ:

## TELEGRAM:


РАЗРАБОТЧИК НА **PYTHON**

Email: vadimkrupinin@gmail.com
Telegram: @KVadim_K
Github: https://github.com/KVadim-K