NewsBot

DANIIL
MIADZVETSKI

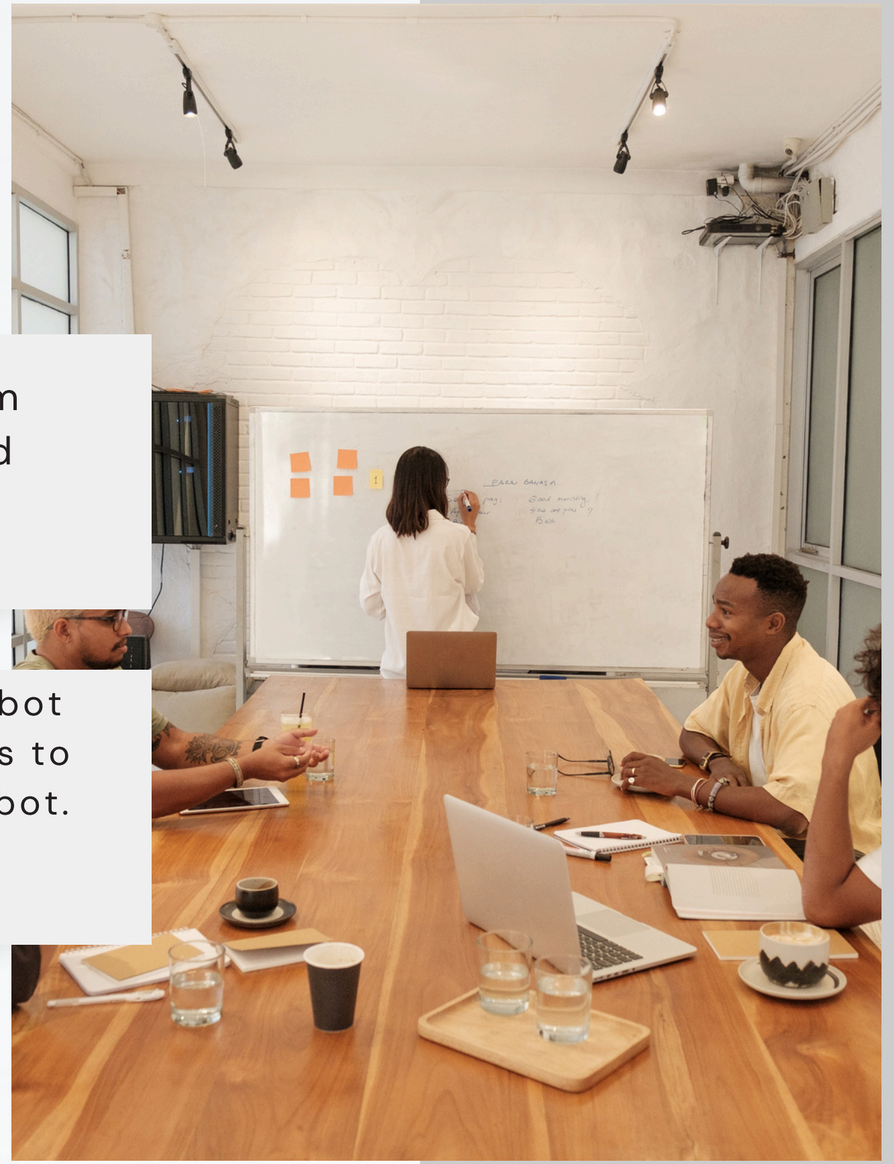
TELEGRAM NEWS BOT FOR WEBSITE

CONTENT

ABOUT ME

I am Daniil Medvetsky, 18 years old, and I am passionate about programming, cycling, and gaming.

The project is a system consisting of a Telegram bot and a Flask-based web application. It allows users to add, edit, and delete news through the Telegram bot. The news is then displayed on the website.



TARGET AUDIENCE

Simplicity

Website owners and administrators who want a simplified way to manage content.

Versatility

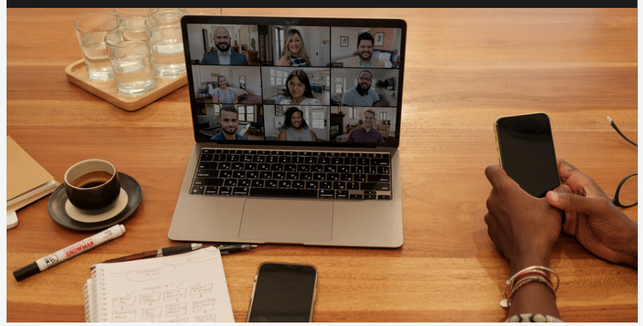
Telegram users who need to quickly publish news without direct access to a website.

Efficiency

Companies and organizations using Telegram for internal communication and information dissemination.

GOALS AND CAPABILITIES OF THE PROJECT

Цели



Simplifying News Publication:

- Provide users with a simple and convenient way to add, edit, and delete news through the Telegram messenger.

Channel and Web Expansion:

- Enable users to expand their Telegram channel's reach to the web, thereby promoting and growing their channel effectively.

Adding News:

- Users can submit news through the Telegram bot, including the headline, description, text, and photos.

Editing and Deleting:

- People have the ability to edit and delete published news directly via Telegram.

Displaying News:

- News added through Telegram is automatically displayed on the website, ensuring up-to-date content for users.

Възможности

MODULES FOR DEVELOPMENT AND FUNCTIONS

- Receives user commands (start, add news, edit news, delete news).
- Collects information about the news (title, photo, description, text).
- Sends data to the server for adding, editing, and deleting news.

Library: pyTelegramBotAPI

- Processes data received from the Telegram bot.
- Manages news (adding, editing, deleting) with storage in the SQLite database.
- Displays news on the website.

LIBRARY: FLASK

Stores information about the news, including:

- Identifier
- Title
- Description
- Text
- Photo path

sqlite3

KEY CODE COMPONENTS

```python
@bot.message_handler(commands=['start'])
def start_message(message):
    markup = types.ReplyKeyboardMarkup()
    btn_1 = types.KeyboardButton('Добавление новости')
    btn_2 = types.KeyboardButton('Редактирование новости')
    btn_3 = types.KeyboardButton('Удаление новости')
    markup.row( *args: btn_1, btn_2, btn_3)
    bot.send_message(message.chat.id, text: 'Здравствуйте! Я бот дл
```

```python
@bot.message_handler(func=lambda message: True)
def handle_message(message):
    user_text = message.text
    if user_text.lower() == "добавление новости":
        bot.send_message(message.chat.id, text: "Введите название новости:")
        bot.register_next_step_handler(message, get_news_title)
    elif user_text.lower() == "редактирование новости":
        bot.send_message(message.chat.id, text: "Введите ID новости, которую нужно отредактировать:")
        bot.register_next_step_handler(message, edit_news)
    elif user_text.lower() == "удаление новости":
        bot.send_message(message.chat.id, text: "Введите ID новости, которую нужно удалить:")
        bot.register_next_step_handler(message, delete_news)
    else:
        bot.reply_to(message, text: f"Принято: {user_text}")
```

```python
def get_news_title(message):
    news_title = message.text
    bot.send_message(message.chat.id, text: "Отправьте фото для новости:")
    bot.register_next_step_handler(message, get_news_photo, *args: news_title)
```

KEY CODE COMPONENTS

```python
def get_news_photo(message, news_title):
    if message.photo:
        file_info = bot.get_file(message.photo[-1].file_id)
        photo = bot.download_file(file_info.file_path)
        bot.send_message(message.chat.id, text: "Введите краткое описание новости:")
        bot.register_next_step_handler(message, get_news_description, *args: news_title, photo)
    else:
        bot.send_message(message.chat.id, text: "Пожалуйста, отправьте фото.")
        bot.register_next_step_handler(message, get_news_photo, *args: news_title)
```

```python
1 usage
def get_news_text(message, news_title, photo, news_description):
    news_text = message.text
    news_id = ''.join(random.choices(string.ascii_letters + string.digits, k=8))
    response = requests.post(flask_server_url + '/add_news', data={'id': news_id, 'title': news_title,
        'description': news_description, 'text': news_text}, files={'photo': photo})
    if response.status_code == 204:
        bot.send_message(message.chat.id, text: f"Новость добавлена с ID: {news_id}")
    else:
        bot.send_message(message.chat.id, text: "Произошла ошибка при добавлении новости")
```

```python
import sqlite3

def init_db():
    conn = sqlite3.connect('news.db')
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS news (
            id TEXT PRIMARY KEY,
            title TEXT,
            description TEXT,
            text TEXT,
            photo_filename TEXT
        )
    ''')
    conn.commit()
    conn.close()
```

КЛЮЧЕВЫЕ ЧАСТИ КОДА

```python
@app.route( rule: '/add_news', methods=['POST'])
def add_news():
    if request.method == 'POST':
        news_id = request.form['id']
        news_title = request.form['title']
        news_description = request.form['description']
        news_text = request.form['text']
        photo = request.files['photo']
        photo_filename = f"{news_id}_{photo.filename}"
        photo.save(os.path.join(app.config['UPLOAD_FOLDER'], photo_filename))

        conn = sqlite3.connect('news.db')
        cursor = conn.cursor()
        cursor.execute( __sql: 'INSERT INTO news (id, title, description, text, photo_filename) VALUES (?, ?, ?, ?, ?)',
                        __parameters: (news_id, news_title, news_description, news_text, photo_filename))
        conn.commit()
        conn.close()

        return '', 204
```

KEY CODE COMPONENTS

```
    <div class="site-news">
        {% for news in news_list %}
        <div class="news">
            <a href="{{ url_for('news_detail', news_id=news.id) }}" class="news-link">
                <div class="photo">
                    <img src="{{ url_for('static', filename='uploads/' + news.photo_filename) }}" alt="News Photo">
                </div>
                <div class="content">
                    <h3>{{ news.title }}</h3>
                    <p>{{ news.description }}</p>
                </div>
            </a>
        </div>
        {% endfor %}
    </div>
</div>
```

THANK YOU FOR YOUR ATTENTION

Stay up-to-date with instant news:
Telegram Bot for instant publishing!

NewsBot