

## Aplikacja do głosowania online (REDIS)

### 1. Cel projektu

Celem projektu jest stworzenie prostej aplikacji umożliwiającej:

- tworzenie ankiet online,
- głosowanie w ankietach jednokrotnego i wielokrotnego wyboru,
- przeglądanie wyników ankiet (również po ich zakończeniu) w formie liczbowej i graficznej

### 2. Wykorzystane technologie

- Python - implementacja
- Flask - framework webowy
- Redis - baza danych NoSQL
- HTML5, CSS, JavaScript - frontend

### 3. Baza Redis

W ramach projektu zdecydowano się na wykorzystanie bazy danych Redis, należącej do grupy baz NoSQL, zamiast klasycznej relacyjnej bazy danych SQL. Wybór ten wynikał z celów dydaktycznych przedmiotu *Bazy NoSQL*.

Redis jest bazą danych typu klucz–wartość, w której dane przechowywane są w pamięci operacyjnej (RAM). Takie podejście pozwala na bardzo szybki dostęp do danych oraz dużą elastyczność w projektowaniu struktury przechowywanych informacji.

### 4. Model danych w Redis

polls:all	Set	zbiór identyfikatorów wszystkich ankiet
poll:{id}:question	String	treść pytania ankiety
poll:{id}:max-choices	String (int)	maksymalna liczba odpowiedzi możliwych do zaznaczenia
poll:{id}:end_time	String (timestamp)	czas zakończenia ankiety
poll:{id}:options	Hash	możliwe odpowiedzi i liczba oddanych na nie głosów
poll:{id}:voters	Set	zbiór adresów IP, które oddały głos

```
127.0.0.1:6379> SMEMBERS polls:all
1) "1"
2) "2"
3) "3"
4) "4"
127.0.0.1:6379> GET poll:1:question
"Co lubisz je\xc5\x9b\xc4\x87?"
127.0.0.1:6379> GET poll:1:max_choices
"1"
127.0.0.1:6379> HGETALL poll:1:options
1) "Pizze"
2) "1"
3) "Burgery"
4) "0"
5) "Ryby"
6) "0"
7) "Owoce"
8) "0"
127.0.0.1:6379> SMEMBERS poll:1:voters
1) "127.0.0.1"
127.0.0.1:6379>
```

Widzimy, że aktualnie w bazie są 4 ankiety, a także jakie dane są zapisane dla ankiety *id=1*.

## 5. Realizowane operacje

### 5.1 Tworzenie ankiety

- generowanie nowego ID ankiety (w jaki sposób)
- zapis danych ankiety do Redis (*SET, HSET, SADD*)

### 5.2 Pobieranie listy ankiet

- pobieranie wszystkich ID ankiet (*SMEMBERS*)
- odczyt szczegółów każdej ankiety (*GET*)
- dynamiczne określenie statusu ankiety (aktywna/zakończona)

### 5.3 Głosowanie

- walidacja czasu zakończenia ankiety
- walidacja liczby wybranych odpowiedzi
- inkrementacja liczby głosów (*HINCRBY*)
- zapis informacji o oddanym głosie (*SADD*)

### 5.4 Odczyt wyników

- pobranie wszystkich opcji i liczby głosów (*HGETALL*)
- prezentacja danych w postaci listy oraz wykresu kołowego

## 6. Funkcjonalności aplikacji wraz z fragmentami kodu

### 6.1 Lista ankiet

Po wejściu na stronę główną użytkownik widzi listę wszystkich dostępnych oraz zakończonych ankiet wraz z informacją o czasie zakończenia.

Operacje Redis: *SMEMBERS, GET*

# Głosowanie online

[Stwórz nową ankietę](#)

## Lista ankiet

### Dlaczego?

Status: ended, kończy się: 2026-01-20 11:03:00

[Zakończona – zobacz wyniki](#)

### Jakie jest twoje ulubione zwierzę?

Status: active, kończy się: 2026-02-13 18:39:00

[Głosuj](#)

### Co lubisz jeść?

Status: active, kończy się: 2026-02-28 15:21:00

[Głosuj](#)

```
#lista ankiet
@app.route("/polls", methods=["GET"])
def list_polls():
    all_ids = r.smembers("polls:all")
    polls = []

    now = time.time()

    for poll_id in all_ids:
        question = r.get(f"poll:{poll_id}:question")
        end_time = float(r.get(f"poll:{poll_id}:end_time"))
        max_choices = int(r.get(f"poll:{poll_id}:max_choices"))

        status = "active" if end_time > now else "ended"
        end_datetime = datetime.fromtimestamp(end_time).strftime("%Y-%m-%d %H:%M:%S")

        polls.append({
            "id": poll_id,
            "question": question,
            "max_choices": max_choices,
            "end_time": end_datetime,
            "status": status
        })

    polls.sort(key=lambda x: x["end_time"])
    return jsonify(polls)
```

## 6.2 Tworzenie nowej ankiety

Użytkownik może utworzyć nową ankietę, podając treść pytania, listę odpowiedzi, maksymalną liczbę zaznaczonych odpowiedzi oraz czas zakończenia ankiety za pomocą kalendarza.

Id ankiety jest generowane automatycznie jako kolejna wolna liczba

Operacje Redis: *SET, HSET, SADD*

# Głosowanie online

## Nowa ankieta

Pytanie:

Opcje (oddzielone przecinkiem):

Maksymalna liczba odpowiedzi:

Czas zakończenia: dd.mm.yyyy --:--

```

#tworzenie ankiety
@app.route("/poll", methods=["POST"])
def create_poll():
    data = request.json

    #id
    all_ids = r.smembers("polls:all")
    if all_ids:
        max_id = max(int(pid) for pid in all_ids)
        poll_id = str(max_id + 1)
    else:
        poll_id = "1"

    question = data["question"]
    options = data["options"]
    max_choices = int(data["max_choices"])
    end_time = data["end_time"]

    r.set(f"poll:{poll_id}:question", question)
    r.set(f"poll:{poll_id}:max_choices", max_choices)
    r.set(f"poll:{poll_id}:end_time", end_time)

    for option in options:
        r.hset(f"poll:{poll_id}:options", option, 0)

    r.sadd("polls:all", poll_id)

    return jsonify({"status": "Poll created", "id": poll_id}), 201

```

### 6.3 Głosowanie w ankiecie

Po wybraniu ankiety użytkownik przechodzi na stronę głosowania i może zaznaczyć maksymalnie tyle odpowiedzi, ile jest podane w *max\_choices*.

System blokuje wielokrotne głosowanie jednego użytkownika dzięki zapamiętywaniu adresu IP przy oddawaniu głosu.

Operacje Redis: *SISMEMBER*, *HINCRBY*, *SADD*

# Głosowanie online

[Powrót do listy ankiet](#)

## Jakie jest twoje ulubione zwierzę?

- kot
- królik
- pies

[Oddaj głos](#)

```
#głosowanie
@app.route("/vote", methods=[ "POST"])
def vote():
    data = request.json
    poll_id = data["poll_id"]
    selected_options = data["options"]
    voter_id = request.remote_addr

    end_time = r.get(f"poll:{poll_id}:end_time")
    if end_time is None or time.time() > float(end_time):
        return jsonify({"error": "Poll has ended"}), 403

    voters_key = f"poll:{poll_id}:voters"
    if r.sismember(voters_key, voter_id):
        return jsonify({"error": "You have already voted"}), 403

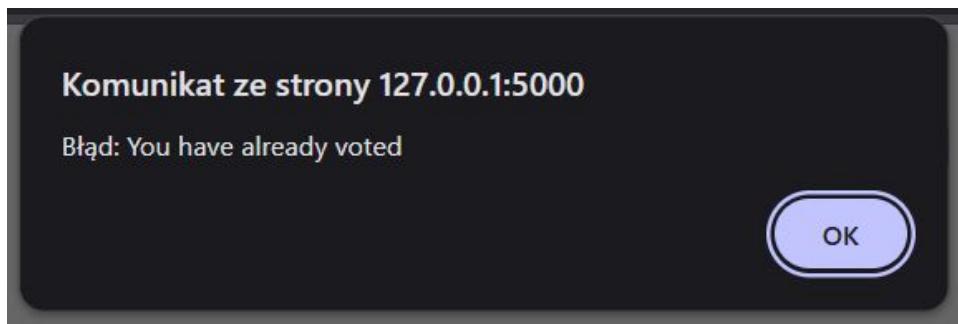
    max_choices = int(r.get(f"poll:{poll_id}:max_choices"))

    if len(selected_options) == 0 or len(selected_options) > max_choices:
        return jsonify({"error": f"You can choose from 1 to {max_choices} options"}), 400

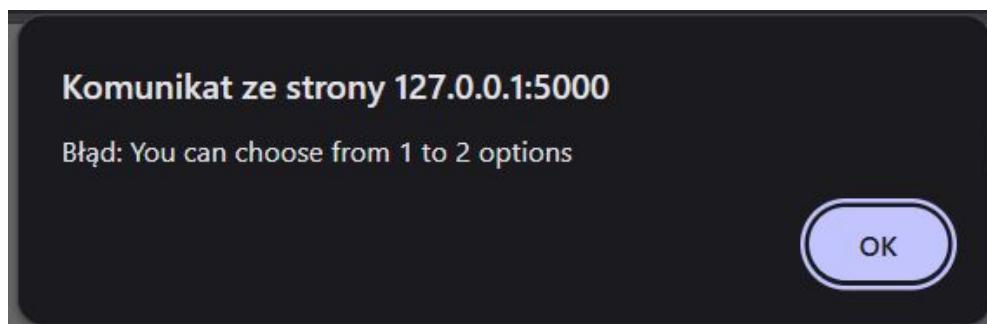
    for option in selected_options:
        r.hincrby(f"poll:{poll_id}:options", option, 1)

    r.sadd(voters_key, voter_id)
    return jsonify({"status": "Vote counted"}), 200
```

W przypadku próby oddania głosu ponownie, system blokuje tę akcję



W przypadku zaznaczeniu zbyt wielu odpowiedzi, system blokuje oddanie głosu i wyświetla stosowny komunikat



#### 6.4 Przeglądanie wyników ankiety

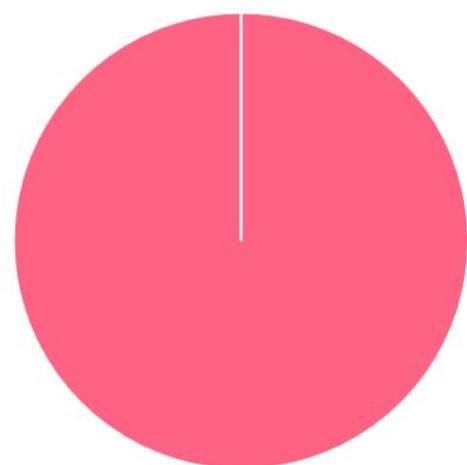
Wyniki ankiety są dostępne zarówno w trakcie jej trwania, jak i po jej zakończeniu. Prezentowane są w formie liczbowej oraz graficznej.

Operacje Redis: *HGETALL*

### Wyniki

- kot: 1 głosów
- królik: 0 głosów
- pies: 0 głosów

kot królik pies



```
#wyniki ankiety
@app.route("/results/<poll_id>", methods=[ "GET"])
def results(poll_id):
    question = r.get(f"poll:{poll_id}:question")
    max_choices = r.get(f"poll:{poll_id}:max_choices")
    end_time = r.get(f"poll:{poll_id}:end_time")
    options = r.hgetall(f"poll:{poll_id}:options")

    return jsonify({
        "question": question,
        "max_choices": max_choices,
        "end_time": end_time,
        "results": options
    })
```