



Fundusze Europejskie
dla Wielkopolski



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



SAMORZĄD
WOJEWÓDZTWA
WIELKOPOLSKIEGO

Załącznik nr 1 do OPZ

**WYTYCZNE DLA WYKONAWCY W ZAKRESIE UTWORZENIA STRONY, MECHANIZMU GŁOSOWANIA
I WERYFIKACJI GŁOSÓW W PLEBISCYCIE „ŁĄCZY NAS WIELKOPOLSKIE”**

NA POTRZEBY ZAMÓWIENIA:

**Przygotowanie, zredagowanie, dystrybucja i promocja 26 wydań E-magazynu „Nasz Region” oraz
organizacja trzech edycji plebiscytu „Łączy nas WIELkopolskie”**

DPR-III-4-1.434.4.5.2023



Rozdział I

Stworzenie strony internetowej Plebiscytu

1. Wykonawca przed przystąpieniem do utworzenia strony Plebiscytu zaproponuje Zamawiającemu trzy wersje wyglądu strony, która swoimi layoutem i funkcjonalnościami będzie nawiązywała do istniejącej już strony laczynaswielkopolskie.pl. Zamawiający może zgłosić uwagi do projektów lub zaakceptować jedną z propozycji. Termin akceptacji/poprawek tego zadania musi zakończyć się w ciągu 5 dni od otrzymania przez Zamawiającego trzech wersji.
2. Zamawiający zapewni Wykonawcy wszelkie niezbędne dostępy związane z hostingiem (panel administracyjny hostingu) oraz wszystkie inne dostępy powiązane, niezbędne do wykonania zadań.
3. Zarejestrowanie odpowiedniej nazwy domeny i przekierowanie adresu DNS na serwery tego hostingu (akcja wykonywana w panelu administracyjnym hostingu) leży po stronie Zamawiającego.
4. Wykonawca ma obowiązek stworzenia co najmniej jednego środowiska testowego:
 - 1) stworzenia subdomeny typu dev/staging np.
 - dev.laczynaswielkopolskie.pl
 - staging.laczynaswielkopolskie.pl
 - 2) Zadbana o tzw. **basic auth**, czyli monit z użytkownikiem/hasłem, który ogranicza grono osób, które mogą odwiedzić dany testowy adres.
5. Wykonawca wybierze Framework'a typu **CMS**, na którym oparta będzie strona Plebiscytu (równoważny z Wordpress w wersji 6.4.2)
6. Wykonawca upewni się, czy dany hosting zapewni niezbędne wymagania dla wybranego Frameworku. Np. dla Wordpress wymogi są następujące (<https://developer.wordpress.org/advanced-administration/before-install> lub <https://wordpress.org/about/requirements/>):
 - 1) Język **PHP** w wersji 7.4 bądź równoważnej.
 - 2) Baza danych **MySQL** w wersji 5.7 lub równoważnej, albo **MariaDb** w wersji 10.4 lub równoważnej.
 - 3) Obsługa protokołu **HTTPS** (wygenerowanie certyfikatu **SSL** na hostingu dla domeny) - koszty powinien ponosić Zamawiający.
7. Stworzenie bazy danych **SQL**, wybranie **prefixu** dla tabel w skrypcie instalacyjnym oraz stworzenie użytkownika bazy danych typu **root**.
8. Zainstalowanie Framework'a przez skrypt instalacyjny dostępny na danym typie hostingu, bądź też przez bezpośredni dostęp po **ssh**.
9. Konfiguracja bazy danych we Framework'u przez podpięcie danych dostępowych serwisu w plikach konfiguracyjnych.
10. Powtórzenie punktów 6 do 8 także na środowisku/środowiskach testowych.
11. Stworzenie konta pocztowego na danym hostingu, które będzie podpięte pod serwis.



Rozdział II

Utrzymanie strony internetowej Plebiscytu

1. Zamawiający będzie właścicielem hostingu, domeny, repozytorium kodu oraz posiadał wszystkie niezbędne dostępy typu **Administrator**, również po ustąpieniu współpracy z Wykonawcą.
2. Wykonawca ustali co najmniej dwa kanały komunikacji z Zamawiającym (np. e-mail oraz kanał komunikacyjny typu **RocketChat, Slack**)
3. Zamawiający zapewni Wykonawcy dostęp niezbędny do utrzymania serwisu:
 - 1) Dostęp do panelu administracji hostingiem (link, użytkownik, hasło),
 - 2) Dostęp do bazy danych Framework'a (adres IP/link do [phpMyAdmin](#), użytkownik typu **root**, hasło),
 - 3) Link do Panelu Admina serwisu, oraz użytkownik, hasło dla Administratora,
 - 4) Dostęp po **ssh** (uprawnienia, adres ip, numer portu) do fizycznej lokalizacji serwisu (do serwera, do katalogu głównego projektu)
 - 5) Dostęp **sftp** (użytkownik, hasło) do katalogu głównego projektu
 - 6) Dostęp do poczty, konta pocztowego podpiętego do serwisu
 - 7) Dostęp do repozytorium kodu projektu (np. **Github, Gitlab** bądź **Bitbucket**)
4. Wymagane jest zebranie i przechowanie kompletnych danych dostępowych – pierwsze podejście
 - 1) głównym Administratorem, wraz ze wszystkimi uprawnieniami jest Zamawiający
 - 2) wszystkie dane dostępowe zebrane w jednym arkuszu kalkulacyjnym, zabezpieczonym hasłem.
 - 3) Przekazywanie pliku poprzez jeden z kanałów informacyjnych np. e-mail, a hasło drugim kanałem informacyjnym (**Rocket Chat, Slack**, bądź **SMSem**)
5. Wymagane jest utworzenie Menadżera haseł, czyli zamiast pliku, serwis typu [LastPass](#), [Bitwarden](#) – drugie podejście
 - 1) dostępy można współdzielić, udostępniać innym użytkownikom w zależności od uprawnień, użytkownik może używać haseł nawet bez ich jawnego odsłonięcia
 - 2) można tworzyć grupy i wokół grup zbierać wszystkie dostępy
6. Dopuszcza się dwa sposoby tworzenia treści/funkcjonalności serwisu:
 - 1) Developerski – przez ingerowanie w kod aplikacji i umieszczanie go na serwerze (deploy)
 - a) Stworzenie repozytorium projektu na **Github, Gitlab** lub **Bitbucket**
 - b) Stworzyć skrypty przeprowadzające deploy. Mechanizm **CI/CD** - opcjonalnie
 - 2) Edytorski – tworzenie treści/stron przez użytkownika panelu Admina.

Rozdział III

Agregowanie treści archiwalnych z poprzednich edycji

1. Przy użyciu wybranego Framework'a Wykonawca zorganizuje i zarchiwizuje poprzednie edycje plebiscytu ze starej strony <https://www.laczynaswielkopolskie.pl/>
2. Wykonawca utworzy kategorię dla każdej edycji oraz przypisze tę kategorię do wszystkich edycji.
3. Wykonawca utworzy niestandardowy typ wpisu jeżeli będzie to potrzebne, aby zachować unikalną strukturę poprzednich edycji plebiscytu, z własnym zestawem podkategorii lub taksonomii.



4. Wykonawca doda artykuły jako podsekcje (zachowa logiczny podział poprzednich edycji, jak było na poprzedniej stronie www):
 - 1) Każda edycja składa się z:
 - a) kilku artykułów
 - b) z opisu wszystkich Kategorii
 - c) z opisu wszystkich nominowanych (Beneficjentów)
 - d) Prezentowanych wyników głosowania, które zostały ostatecznie zweryfikowane i uznane za oficjalne
 - 2) Wykonawca wykorzysta nagłówki, akapity i inne narzędzia formatowania do organizacji artykułów w obrębie sekcji danej edycji plebiscytu.
5. Wykonawca utworzy stronę Archiwum (wszystkie edycje plebiscytu według lat).
6. Wykonawca użyje tagów dla motywów lub tematów.
7. Wykonawca wdroży nawigacyjne Menu, w którym umieści linki do głównej strony poprzednich edycji plebiscytów i do ewentualnych konkretnych kategorii lub do archiwa tagów.
8. Wykonawca będzie promował Archiwum na stronie internetowej: doda linki do archiwum poprzednich edycji plebiscytu w nawigacji lub stopce strony internetowej.
9. Wykonawca wyeksponuje obecną edycję plebiscytu na pierwszym miejscu serwisu, jako centralne miejsce.

Rozdział IV

Stworzenie mechanizmu Plebiscytu i weryfikacja wyników

Wymagania ogólne

1. Wykonawca musi wziąć pod uwagę, że plebiscyt „Łączy nas WIELkopolskiE” jest skierowany do wszystkich osób chcących wziąć udział w głosowaniu, bez względu na wiek, miejsce zamieszkania czy narodowość, choć w pierwszej kolejności jest ono kierowane do mieszkańców Wielkopolski. Wykonawca zapewni prostotę i przejrzystość oraz intuicyjność wszystkich mechanizmów umożliwiających oddanie głosu.
2. Głosowanie dotyczy 20 nominowanych projektów. Za każdym projektem stoi dany beneficjent. Projekty rozdzielone są równo pomiędzy 4 kategorie. W każdej kategorii jest zestaw 5-ciu projektów. Można w danej kategorii spośród wspomnianych wyżej 5 projektów oddać tylko 1 głos w danym dniu. Można oddać maksymalnie 4 głosy dziennie (bo tyle jest dostępnych kategorii). Przy czym należy uwzględnić, że liczba projektów, kategorii oraz dane projektów/beneficjentów mogą ulegać zmianie zanim nastąpi uruchomienie plebiscytu. Moduł/plugin, o którym mowa, powinien umożliwiać wprowadzanie takich zmian w sposób intuicyjny i łatwy.
3. Wykonawca na stronie plebiscytu, zamieści aktualne, widoczne wyniki głosowania w czasie trwania plebiscytu, w poszczególnych kategoriach (histogram procentowy lub liczbowy, do decyzji Zamawiającego).



4. Główne kryteria oraz wymagania modułu/pluginu umożliwiającego przeprowadzenie tego plebiscytu oraz weryfikację uzyskanych przy jego pomocy wyników, które muszą być spełnione to:

1) Cyfrowe bezpieczeństwo dla użytkownika biorącego udział w głosowaniu.

- a) Dane wrażliwe są zabezpieczone, w ten sposób, że nie są zbierane, nie są przekazywane podmiotom trzecim ani nigdzie nie są zapisywane.
- b) Do bazy danych nie jest zapisywany e-mail użytkownika w postaci jawnej. Jest wykorzystywany jedynie do prostej weryfikacji użytkownika:
 - Zabezpieczenie wszystkich używanych formularzy mechanizmem chroniącym przed atakami typu **CSRF** czy **SQL Injection**
 - Poddanie testowi Turinga głosującego w czasie wypełniania formularzy (zastosowanie **ReCaptcha V2 Invisible** lub lepszej).
 - Weryfikacja zgodności 6 cyfrowego hasła **OTP** przesyłanego na e-mail, który potwierdza głosujący przepisując go z wiadomości e-mail do formularza. Po zatwierdzeniu formularza przez głosującego następuje porównanie numeru zapisanego w sesji na serwerze z tym wpisanym przez niego. Na podstawie porównania tych numerów następuje autoryzacja bądź nie, w zależności od wyniku.
- c) Użytkownik ma łatwą możliwość zapoznania się z regulaminem udziału w plebiscycie (kliknięcie w link obok zgody).
- d) Użytkownik musi wypełnić pole z imieniem/nickiem oraz e-mailem, ale nazwisko jest opcjonalne.

2) Cyfrowe bezpieczeństwo podmiotu przeprowadzającego głosowanie.

- a) Zapewnienie przez moduł/plugin weryfikacji, że za każdym oddanym głosem stoi prawdziwy człowiek, który intencjonalnie i dobrowolnie wziął udział w głosowaniu.
 - Głosujący, musi zaznaczyć zgodę, że zapoznał się z zasadami głosowania (konieczna zgoda typu „zapoznałem się z regulaminem udziału w plebiscycie”).
 - Formularz musi posiadać walidację frontową dotyczącą pól wymaganych (imię/nick, e-mail oraz wybór zgody z zapoznaniem się z regulaminem)
 - Formularz musi posiadać walidację backendową. Do walidowanych pól wymienionych w punkcie 2.1.2 dochodzi też pole *project_id*, które oprócz tego, że: jest obecne w przesłanych danych; jest liczbą większą od zera; to musi być też prawdziwym id projektu, czyli realnie musi istnieć w bazie.

Uwaga: Pole `project_id` może być zapisane jako atrybut (a także `category_id`) na przycisku do głosowania.

- Formularz musi być zabezpieczony przed atakami **CSRF** oraz **SQL Injection**
- Formularz musi być zabezpieczony **ReCaptcha V2 Invisible** bądź lepszą



- Formularz musi być 2 stopniowy wymagający od głosującego przepisania bezbłędnie z maila kodu **OTP**

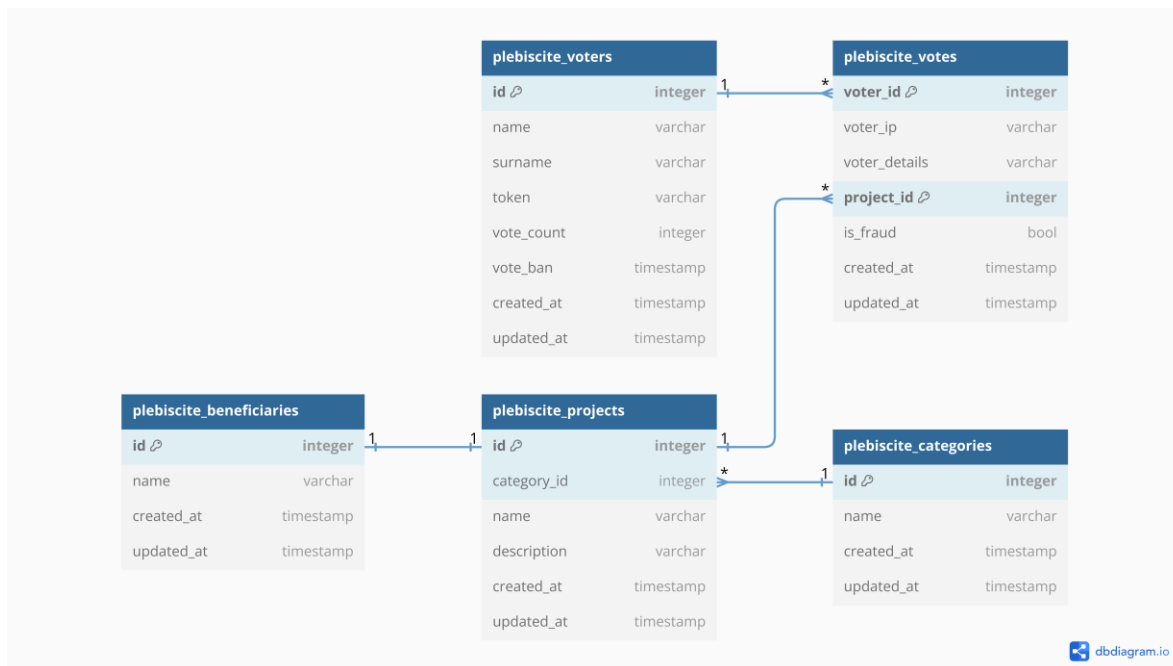
- b) W każdym innym przypadku próba oddania głosu (zapisania w bazie danych) jest uniemożliwiona przez mechanizmy zabezpieczeń rzeczzonego modułu/pluginu. Warto też zwrócić uwagę że przemyślane zaprojektowanie bazy danych może ten proces również ułatwić.
- c) Wykonawca zapewni drugie źródło weryfikacji uzyskanych wyników, uzyskanych w sposób alternatywny, przy pomocy innych dostępnych metryk takich jak eventy **Google Tag Manager'a** czy **Google Analytics**, przy czym wciąż trzeba mieć na myśli bezpieczeństwo cyfrowe głosującego jak i podmiotu przeprowadzającego plebiscyt – (chodzi o rodzaj przesyłanych tam danych)

Drugie źródło umożliwi porównanie, uwiarygodnienie wyników. Może się okazać pomocne, w sytuacji, gdy Zamawiający podczas plebiscytu będzie się mierzył z nieuczciwymi próbami przekroczenia regulaminu plebiscytu.

- 3) Mierzalność, możliwa interpretacja ilościowa i jakościowa uzyskanych wyników: tworzenie raportów cząstkowych oraz stworzenie raportu całościowego.
 - a) Zaprojektowanie bazy danych za wytycznymi podanymi w dalszej części.
 - b) Stworzenie raportu z uzyskanych wyników, który umożliwi ogłoszenie wyników plebiscytu zaraz po jego zakończeniu.
 - c) Porównanie wyników zebranych w module/pluginie z wynikami zebranymi w systemach zewnętrznych (eventy **GTM, GA**)
 - d) Zapewnienie możliwości podmiotowi przeprowadzającego plebiscyt w prosty sposób tworzenia raportów cząstkowych, szacunkowych w czasie trwania całego plebiscytu oraz podglądu bieżących wyników głosowania.
- 4) Spełnienie wszystkich pozostałych założeń i oczekiwań podmiotu co do sposobu przeprowadzenia plebiscytu.
- 5) Wykonawca weźmie pod uwagę daty rozpoczęcia i zakończenia plebiscytu, to znaczy, że moduł/plugin może wyświetlać formularz z głosowaniem, tylko w tym określonym przedziale czasowym. Zestaw dat, treść obu tekstów dostarczy Zamawiający.

Wymagania szczegółowe modułu/pluginu

1. Projekt bazy danych powinien obejmować minimum 4 tabele powiązane ze sobą kluczami obcymi. Nie musi (a nawet nie może), ze względu na założenia plebiscytu występować unikalność pary *voter_id* oraz *project_id* w tabeli zbierającej głosy (zatem można wprowadzić *id*, jako klucz główny dla tabeli *plebiscite_votes*).



2. Moduł/plugin powinien zawierać część administracyjną dla podmiotu zlecającego, oraz frontową dla głosującego.
3. W części administracyjnej
 - 1) Konfiguracja powinna zawierać następujące elementy:
 - a) Main settings:
 - *description* – opis wyświetlany na stronie głosowania
 - *schedule* – wybór wyświetlania ciągłego plebiscytu/okresowego
 - *start_time* – jeżeli okresowego to data zaczęcia plebiscytu
 - *stop_time* – jeżeli okresowego to data zakończenia plebiscytu
 - *max_votes* – ile razy użytkownik może zagłosować w przeciągu całego plebiscytu
 - *nonvoting_period* – jak długo ma trwać zakaz ponownego głosowania (domyślnie 24 h)
 - *before_html* – html wyświetlany przed rozpoczęciem plebiscytu na stronie plebiscytu
 - *after_html* – html wyświetlany po zakończeniu plebiscytu na stronie plebiscytu
 1. *rules_and_regulations* - tekst regulaminu, o którym była mowa we wcześniejszych punktach
 - b) ReCaptcha:
 - *sitekey* – klucz publiczny ReCaptcha



- *secretkey* – klucz prywatny ReCaptcha (o ile to możliwe wyświetlany jak hasło kropki zamiast widocznych znaków)

c) Emails:

- *name* – nazwa nadawcy maila z kodem **One Time Password**
- *email* – email nadawcy maila z kodem **OTP**
- *email_template* – szablon mailowy z kodem **OTP**

d) GTM Events:

- *name1* – nazwa eventu (gdy głosujący klika na przycisk zagłosuj (dane przesyłane to *project_id* i *category_id*))

- *name2* – nazwa eventu (gdy głosujący wypełnia formularz i go zatwierdza, by móc zagłosować (dane przesyłane to *name*, *surname*, *token*, *project_id*))

- *name3* – nazwa eventu (gdy głosujący wysła zatwierdza formularz z przepisany kodem OTP (dane przesyłane to *name*, *surname*, *token*, *project_id*, *status*, *resultText*))

- GTM ID – tutaj wpisuje się id kontenera (w formacie GTM-YYYYYYY)

2) Gridy – tabele do manipulacji nad danymi edytowalnymi dla podmiotu

a) Grid **Categories** - kolumny wyświetlane (ID, Name, Action).

- Przycisk nad gridem Add New Category – dodający nową kategorię

- Action: edycja kategorii,

- Mass actions: Delete – usuwanie zaznaczonych kategorii

b) Grid **Projects** – kolumny wyświetlane (ID, Category, Beneficiary Name, Name, Description, Action)

- Przycisk nad gridem Add New Project – dodający nowy projekt

- Action: edycja projektu

- Mass actions: Delete – usuwanie zaznaczonych projektów

c) Grid **Beneficiaries** – kolumny wyświetlane (ID, Name, Action)

- Brak przycisku nad gridem

- Action: edycja beneficjenta

- Mass actions – brak

d) Grid **Voters** – kolumny wyświetlane (ID, Name, Surname, Token, Vote Count, Vote Ban)

- Brak przycisku nad gridem

- Action – brak

- Mass actions: Delete – usuwanie zaznaczonych użytkowników

e) **Report** - Grid **Votes** – kolumny wyświetlane (ID, Voter ID, Voter IP, Voter Details, Project, Category, Is Fraud, Created At)

- Wykres nad gridem (tworzenie histogramu na podstawie danych przefiltrowanych dla których *is_fraud* = false)



- Mass actions: *Mark Fraud* – wybrane głosy zaznacza jako te, które chcemy odjąć, *Mark Not Fraud* – wybrane głosy odznacza, że to oszustwo w tryb normalny, *Add To Chart* – zaznaczone głosy dodaje do kolekcji, na podstawie, której zostanie wyrysowany histogram (który projekt ile zdobył głosów, posortowany od największej do najmniejszej liczby głosów)

- Filtry. Tutaj uwaga, bardzo ważne, aby w gridzie wyświetlającym

Search by keyword

Filters

Default View

Columns

8 records found (2 selected)

20 per page

1 of 1

Actions

Mark Fraud

Mark Not Fraud

Add To Chart

					Project	Category	Is Fraud	Created At
☒	270	114	172.23.0.13	Cookie: BUKKAPAD7E5MAFE0E3FB8BC062BF... 552360.4633.3450.0542877278h... re: (google/108.163.108.102:443) ... p: ... T: ... C: ... W: ... Leng: ...	Instalacja odnawialnych źródeł energii w gospodarstwach domowych dla mieszkańców Gminy Miejskiej Kościan	Łączy nas... dobry klimat	☒	Jan 29, 2024 11:18:45 PM
☒	269	111	172.23.0.13	re: ... p: ... l: ... C: ... W: ... Leng: ...	Wielkopolskie Centrum Ekonomii Solidarnej	Łączy nas... społeczna misja	☒	Jan 29, 2024 11:17:06 PM

wszystkie głosy działały filtry, które pełnią kluczową rolę w tworzeniu raportów częściowych jak i ostatecznego raportu całego plebiscytu. Oto one:

- *Voter ID* – filtr tekstowy
 - *Voter IP* – filtr tekstowy
 - *Voter Details* – filtr tekstowy
 - *Project* – filtr typu multi select z wszystkimi projektami
 - *Category* – filtr typu multi select z wszystkimi kategoriami
 - *Is Fraud* – filtr typu select
 - *Created At* – zakres dat. Ten filtr jest zwłaszcza istotny, gdyż wszelkie dane głosowania mają zawierać się w określonym czasie.
4. W części frontowej
- 1) W zakładkach poziomych jako etykiety wypisane kategorie, zakładki można przełączać
 - 2) W treści zakładek wypisane projekty wraz z opisami
 - 3) Pod każdym opisem przycisk Zagłosuj (zawiera atrybuty *data-id* (project_id) oraz *cat-data-id* (category_id)), kliknięcie otwiera pop up, na którym znajduje się formularz, który składa się z pól:
 - a) First Name – pole wymagane (imię bądź nick)
 - b) Last Name – pole opcjonalne
 - c) E-mail – pole wymagane
 - d) Check box I confirm I read and agree with rules and regulations (po kliknięciu w ten pseudo link otwiera się pop up na którym wypisany jest regulamin udziału w głosowaniu) – pole wymagane. Treść regulaminu dostarcza Zamawiający.
 - e) Przycisk Send – wysyła pierwszy raz dane. Przycisk zawiera badge **ReCaptcha V2 Invisible**
 - f) Zawiera pole niewidoczne *form_key* od zabezpieczenia **CSRF** (wygenerowany kod na formularzu, oraz w ciastku)
 - g) Zawiera pole niewidoczne *project_id* wybranego projektu



- 4) W czasie pierwszego wystąpienia formularza jest przeprowadzana walidacja
 - a) pola *form_key* – przeciwko atakowi **CSRF**
 - b) sprawdzany nagłówek żądania *X_REQUESTED_WITH* czy zawiera wartość *XMLHttpRequest* – sprawdzanie, czy żądanie jest typu **AJAX**
 - c) pole *g-recaptcha-response*, odpytanie przez **API** serwisu **ReCaptcha**, czy nie znajduje nic niepożądanego odnośnie tego requestu (wartość IP, wykonywane działania przez 'głosującego' bądź bota)
- 5) Następnie wysyłany jest e-mail na adres podany przez głosującego z wygenerowanym losowo 6 cyfrowym kodem **OTP**, jednocześnie zwracany jest zapytaniem AJAXowym formularz wzbogacony o 6 pól z walidacją ustawioną dla 1 cyfry tylko (można podać tylko 1 cyfrę w każdym polu)
- 6) Głosujący w tym momencie ma przepisać kod z maila i po każdej cyfrze wpisać w te 6 pól i jeszcze raz zatwierdzić formularz. Cały proces walidacji przebiega tak samo jak poprzednio
- 7) Jeżeli kod OTP z maila i przepisany do formularza zgadza się z tym zapamiętanym w sesji na serwerze, następuje kolejny szereg walidacji. Sprawdzane jest czy użytkownik istnieje, jeżeli tak, to czy nie przekroczył liczby dopuszczalnych głosów i czy obecny moment w czasie nie mieści się w czasie *vote_ban*. Jeżeli wszystko przebiega poprawnie, dane głosującego są update'owane (*vote_count* i *vote_ban*) oraz dopisywany jest głos do kolejnego wiersza w bazie danych. Jeżeli użytkownik nie istnieje to jest tworzony w bazie danych, pola *vote_count* i *vote_ban* są uzupełniane i głos też jest dodawany. Głosujący otrzymuje komunikat potwierdzający głosowanie, lub podający przyczynę odrzucenia głosu. Komunikaty, które są wyświetlane głosującemu na koniec głosowania:

'Thank you for your precious vote!'

'Error happened [You cannot vote more times a day than allowed. Please do it later.]'

'Error happened [You have already voted for a project in this category today.]'

- 8) Tworzenie użytkownika: w celu weryfikacji głosującego jest potrzebny adres e-mail. Założeniem modułu/pluginu jest cyfrowe bezpieczeństwo głosującego, zatem adres ten nie jest przechowywany w sposób jawny tylko w zaszyfrowany na polu *token*. Do zaszyfrowania wystarczy format Base64, bądź inny.
- 9) W międzyczasie wysyłane są eventy GTM'a w momencie gdy głosujący klika przycisk Głosuj, gdy zatwierdza formularz ze swoimi danymi pierwszy raz oraz gdy już zatwierdza formularz z przepisaniem kodem OTP w tym 6 polach.

Mierzenie tych 3 eventów pozwoli sprawdzić ilu użytkowników przechodzi cały proces, a ilu odpada z jakiegoś powodu i już nie kończy oddawania głosu.

Dane zbierane tą metodą pozwolą pokazać, czy nie jest używany jakiś automat do głosowania. **ReCaptcha** co jakiś czas każe głosującemu kliknąć w kafelki gdzie są rowery, samochody, mosty czy przejścia dla pieszych. Druga bariera pojawia się przy przepisaniu kodu **OTP**. Poprawne przepisanie tego kodu oznacza poprawny adres e-mail, do którego dostęp ma głosujący. To powinno skutecznie wyeliminować boty, czy inne automaty bądź złośliwe oprogramowanie.

5. Wszystkie informacje na stronie plebiscytu oraz podczas głosowania, o których mowa w niniejszych wytycznych muszą wyświetlać się dla odbiorcy w języku polskim.