



Temat: Projekt Inżynierski – Dokumentacja projektu 'Anonimowa Ankieta'

Autorzy: Mateusz Nosek, Miłosz Kamiński

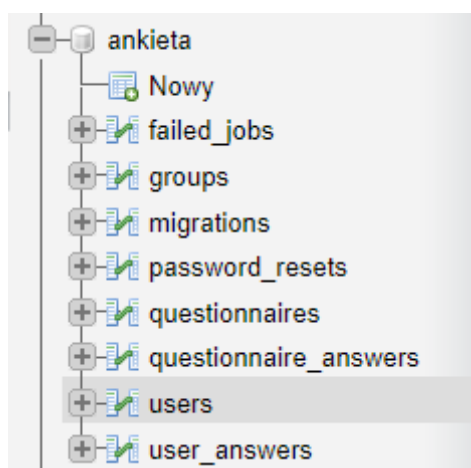
## 1. Opis projektu

Projektem jest system webowy, który umożliwił będzie przeprowadzanie ankiet w sposób gwarantujący anonimowość użytkowników. Aplikacja realizuje następujące funkcje:

- Informacje przechowywane są w odpowiedni sposób w bazie danych
- Reprezentacja danych nie umożliwia na powiązanie użytkownika z konkretnymi danymi
- Istnieje możliwość, aby osoba z wypełnionymi przez siebie ankietami mogły zobaczyć swe odpowiedzi.
- Wykładowcy mają możliwość sprawdzenia odpowiedzi anonimowych i niepowiązanych ankiet.

## 2. Baza danych

Projekt wykorzystuje relacyjną bazę danych MySQL. Poniżej przedstawienie i omówienie poszczególnych tabel.



- Failed Jobs– FAILED JOBS (?)
- Tabela groups – Tabela grup studentów

id	year	name	created_at	updated_at
1	2020	Grupa 2020	2020-06-26 21:00:00	2020-06-26 23:00:00

Tabela groups zawiera następujące pola

Id – unikalne id grupy

Year – rok, w którym student rozpoczął edukację

Name - nazwę grupy

Created\_at - datę mówiącą, kiedy grupa powstała

Updated\_at - datę mówiącą o zmianach wprowadzanych w grupie

c. Tabela Migrations

d. Tabela password\_resets - Resetowanie haseł użytkowników

email	token	created_at
-------	-------	------------

Email – Email na który była prośba o reset hasła

Token – Token wysłany na adres email

Created\_at – Data prośby o reset hasła.

e. Tabela questionnaires – tabela ankiet

id	title	data	created_at	updated_at	group_id
1	Ankieta do wypełnienia	[{"type": "radio", "answers": ["1", "2", "3", "4", ...	2020-06-26 20:53:21	2020-06-26 20:53:21	1

Id – Unikalne Id ankiety

Title – Nazwa ankiety

Data – pytania oraz możliwe odpowiedzi w ankiecie.

Created\_at – data stworzenia ankiety

Updated\_at – Data modyfikacji ankiety

Group\_id – przypisanie do odpowiedniej grupy

f. Tabela questionnaires\_answers – Odpowiedzi ankiet

id	questionnaire_id	answers	created_at	updated_at
1	1	[[{"question": "Jak oceniasz, w skali od 1 do 6, gdz...	2020-06-26 20:24:02	NULL

ID – Unikalne ID zestawu odpowiedzi

Questionnaire\_id – ID ankiety, do której są odpowiedzi

Answers – Odpowiedzi udzielone przez użytkownika

Created\_at – data dodania odpowiedzi

Updated\_at – Data update odpowiedzi

g. Tabela Users – tabela użytkowników

id	name	email	email_verified_at	password	remember_token	created_at	updated_at	is_student	display_name	album_number	group_id
1	marek	marek@marek.pl	NULL	\$2y\$10\$bKwWUMQjGETSbzMYAA84/EUzJIE4C5MIC.vTn.VSz...	NULL	2020-06-26 19:03:11	2020-06-26 19:03:11	1	Student Marek	Album 15	1

ID – Unikalne ID użytkownika

Name – Nazwa użytkownika

Email – Email użytkownika

Email\_verified\_at – Data weryfikacji adresu email (Aktualnie nie używane, pole przygotowane do dalszego rozwoju aplikacji)

Password - hasło użytkownika zaprezentowane w sposób zaszyfrowany

Remember\_token – Pole przygotowane do rozwoju aplikacji

Created\_at – Data stworzenia konta

Updated\_at – data wprowadzenia zmian w profilu użytkownika

Is\_student – pole prezentujące jakim typem konta jest użytkownik (domyślnie 1 lub 0, gdzie 1 to konto studenta, 0 to konto pracownika uniwersytetu)

Display\_name – jest to wartość wyświetlana po zalogowaniu

Album\_number – pole które nie musi być wypełnione - tylko studenci będą mieli wpisany swój numer albumu.

Group\_id – Przypisanie grupy do odpowiedniego użytkownika.

#### h. Tabela user\_answers – odpowiedzi użytkowników

id	user_id	questionnaire_id	answers	created_at	updated_at
7	1	1	nHYbXsb+6fvf+gDq7bOi8qa+cAMWD9Qz2VaaLZu3D+jK9q6TJ2...	2020-06-26 21:25:24	NULL

ID – unikalne id rekordu

User\_id – ID usera który udzielił odpowiedzi

Questionnaire\_id – Id ankiety, na którą były udzielane odpowiedzi

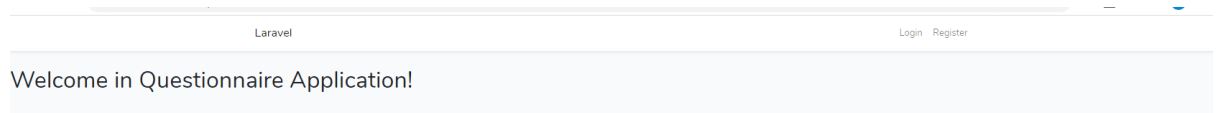
Answers – odpowiedzi użytkownika zaprezentowane w sposób zaszyfrowany – tylko użytkownik ma klucz z możliwością odczytania wiadomości.

Created\_at – data oddania odpowiedzi

Updated\_at – data modyfikacji odpowiedzi (pole stworzone do dalszego rozwoju aplikacji)

Zaprezentowany schemat bazy danych daje możliwość w prosty sposób na przechowanie odpowiedzi studentów w sposób anonimowy jak również pozwala na sprawdzenie odpowiedzi przez użytkownika. Szyfrowanie odpowiedzi, które udzielił użytkownik jest na podstawie klucza podanego przez użytkownika. Dzięki takiemu podejściu odpowiedzi może zobaczyć tylko osoba znająca klucz, który został podany podczas wysyłania odpowiedzi.

### 3. Omówienie aplikacji



Główna strona aplikacji jest prosta i oferuje możliwości takie jak logowanie oraz rejestracja. Po kliknięciu w przycisk 'Register' zostajemy przekierowani na stronę umożliwiającą zarejestrowanie konta.

Register

Name

E-Mail Address

Password

Confirm Password

Register

Aby zarejestrować konto musimy podać dane takie jak nazwę swojego konta, adres email oraz hasło. Rejestracja kończy się dodaniem wpisu do bazy danych do tabeli users.

Przykład:

Dane konta rejestrowanego :

Name: Test

E-Mail Address : [Adrestetstowy@test.pl](mailto:Adrestetstowy@test.pl)

Password: testowehaslo

Register

Name

E-Mail Address

Password

Confirm Password

Register

Po wciśnięciu przycisku 'Register' zostajemy automatycznie zalogowani oraz przekierowani na stronę główną użytkownika.

Laravel Test ▾

Dostępne ankiety do wypełnienia

Wypełnione ankiety

W bazie danych został utworzony rekord z danymi użytkownika. Wygląda to następująco:

Test	Adrestetstowy@test.pl	NULL	\$2y\$10\$1MkH2gqBQ4mrOKGt2gHsu5HzA4czLleor4yFP86zFN...	NULL	2020-07-04 00:02:38	2020-07-04 00:02:38	1	NULL	NULL
------	-----------------------	------	---	------	---------------------	---------------------	---	------	------

Odpowiada za to następujący fragment kodu, który służy do walidacji oraz utworzenia użytkownika

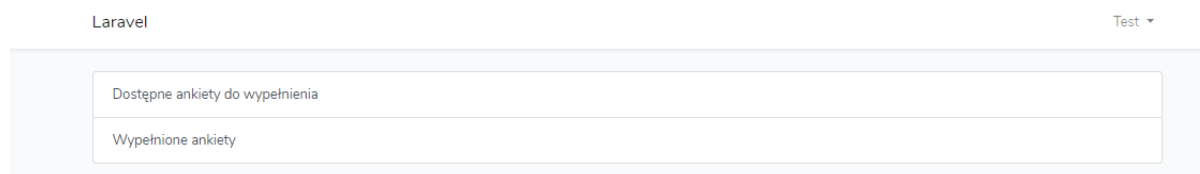
```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}
```

Walidacja sprawdza wszystkie pola, które mamy w zakładce rejestracji. Wszystkie pola są obowiązkowe, pola 'name' oraz 'email' mają ograniczenie znaków - maksymalnie 255 znaków. Hasło użytkownika musi posiadać minimum 8 znaków.

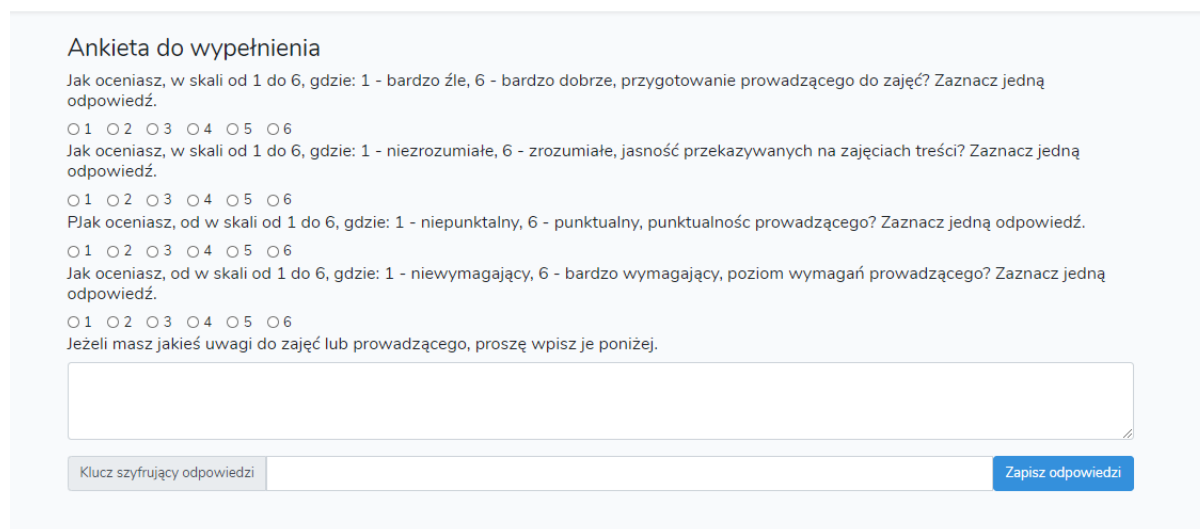
```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
```

Fragment kodu pokazuje stworzenie użytkownika ze wszystkimi danymi podanymi podczas procesu rejestracji. Warunkiem koniecznym przy tworzeniu konta jest poprawne przejście procesu walidacji.

Po zalogowaniu się na konto studenta, pojawia się widok z zakładkami ankiet wypełnionych jak również z ankietami do wypełnienia.



Po wejściu w 'Dostępne Ankiety do wypełnienia' pojawia nam się ankieta którą mamy wypełnić.



Ankieta składa się ze standardowych pytań, na które student udziela odpowiedzi. Pole 'Klucz szyfrujący odpowiedzi' to dodatkowe pole obowiązkowe które ma na celu szyfrować odpowiedzi w zależności od klucza podanego przez użytkownika. Za szyfrowanie odpowiada funkcja w kodzie :



```

final class Cryption
{
    // Store the cipher method
    // https://www.php.net/manual/en/function.openssl-get-cipher-methods.php
    private $ciphering = "AES-128-CTR";

    private $options;
    // Use OpenSSL Encryption method
    private $iv_length ;

    // Non-NULL Initialization Vector for encryption
    private $encryption_iv;

    public function __construct()
    {
        $this->iv_length = openssl_cipher_iv_length($this->ciphering);
        $this->encryption_iv = '1234567891011121';
        $this->options = 0;
    }

    // Use openssl_encrypt() function to encrypt the data
    public function encryption(string $data, string $encryptionKey)
    {
        return openssl_encrypt($data, $this->ciphering, $encryptionKey, $this->options, $this->encryption_iv);
    }
}

```

```

$encryption = new Cryption();
$user = Auth::user();

$string = json_encode($answers);
$str = $encryption->encryption($string, $encryptionKey);

DB::table( table: "user_answers")->insert([
    'user_id' => $user->getAuthIdentifier(),
    'questionnaire_id' => $questionnaireId,
    'answers' => $str,
    'created_at' => date( format: 'Y-m-d H:i:s')
]);

DB::table( table: "questionnaire_answers")->insert([
    'questionnaire_id' => $questionnaireId,
    'answers' => json_encode($answers),
    'created_at' => date( format: 'Y-m-d H:i:s')
]);

return redirect()->route( route: 'questionnaire.answers');

```

Powyższy kod pokazuje jak działa funkcja szyfrująca oraz moment wykorzystania metody szyfrującej w momencie dodawania odpowiedzi do bazy danych. Same odpowiedzi w bazie danych wyglądają następująco:

id	user_id	questionnaire_id	answers	created_at	updated_at
7	1	1	nHYbXsb+6fvf+gDq7bOi8qa+cAMWD9Qz2VaaLZu3D+jK9q6TJ2...	2020-06-26 21:25:24	NULL

answers

text

gFUSq6YF1KquQbKyVQCC9wUYQc+hp61kf4RLa  
wFVNS8P/Daj/fgaGjRoWY2J1vKwFtz2qn11Kd  
vxQZkuJg3N8qfbf8/75c/1eCjLQ5Lp42xUwn3  
VGLwuE8A46f+e/NAX1UUV0gbwZmuIPMD3ZBSH  
NHU9itwB4QsT7p3oeC3capYxKjvpgIm/BcwFb  
1sEkPSY+0NrYzfARRh+4MiFvV3PknabASK48S  
veCw6fwUdm4usxwF1uxsLdJIEPZdhm430c4/  
pYjIipSJAHRP7m11v4KrIyJx06j1XRA4TFXjL  
hpatVh6swI8hJuF558I303w6SEC8S1qawwhsB  
rYVbhkIbpCNnwMqqdb1Dwkm4KqREuvJvX4YJL  
IyUatT1DPXlt7Jb5Q4L+gutUQ0WjrsSD7sQyN  
2vvjS46UCjqwAKtV/b8gIUwPUne2pFbCavo4Y  
Hac768TMpw9e/AXhhB6k1I8vKbStdgA7GTucD  
a38o5SzlIkfly/8BaeqVJfM0rZEA20FvkYJg7  
hj/xP5ftPSEipu5kALGckXyzxTUAf6o1Xwu4j

Odpowiedzi znajdują się w tabeli user\_answers. Dzięki takiemu rozwiązaniu użytkownik, który wypełnił ankietę ma szansę na sprawdzenie swoich odpowiedzi. Jednocześnie nikt nie jest w stanie zidentyfikować osoby która ankietę wypełniła.

Za wyświetlanie dostępnych ankiet do wypełniania odpowiada kod:

```

public function available()
{
    $user = Auth::user();
    $questionnaire = [];
    $availableQuestionnaire = DB::select(DB::raw(
        value: "SELECT a.answers, q.id, q.title, q.data FROM questionnaires q
        JOIN users u ON u.group_id = q.group_id
        LEFT JOIN user_answers a ON q.id = a.questionnaire_id AND a.user_id = u.id where u.id = ".$user->getAuthIdentifier()
    ));

    foreach ($availableQuestionnaire as $item) {
        $questionnaire[] = [
            'id' => $item->id,
            'title' => $item->title,
            'questions' => json_decode($item->data)
        ];
    }

    return view('questionnaire/list', ['questions' => $questionnaire]);
}

```

Po wypełnieniu ankiety przed samym przesłaniem jej użytkownik podaje klucz, który jest znany tylko jemu. Dzięki temu kluczowi możemy odkodować odpowiedzi, które będą widoczne tylko osobie wypełniającą.

Do przeanalizowania zasady działania wypełniania ankiety posłużymy się następującym przykładem:

### Ankieta do wypełnienia

Jak oceniasz, w skali od 1 do 6, gdzie: 1 - bardzo źle, 6 - bardzo dobrze, przygotowanie prowadzącego do zajęć? Zaznacz jedną odpowiedź.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☒ 6

Jak oceniasz, w skali od 1 do 6, gdzie: 1 - niezrozumiałe, 6 - zrozumiałe, jasność przekazywanych na zajęciach treści? Zaznacz jedną odpowiedź.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☒ 6

Płak oceniasz, od w skali od 1 do 6, gdzie: 1 - niepunktualny, 6 - punktualny, punktualność prowadzącego? Zaznacz jedną odpowiedź.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☒ 6

Jak oceniasz, od w skali od 1 do 6, gdzie: 1 - niewymagający, 6 - bardzo wymagający, poziom wymagań prowadzącego? Zaznacz jedną odpowiedź.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☒ 6

Jeżeli masz jakieś uwagi do zajęć lub prowadzącego, proszę wpisz je poniżej.

Testowy

Klucz szyfrujący odpowiedzi

....

Zapisz odpowiedź

W naszym przypadku kluczem szyfrującym jest 'test'.

Po wypełnieniu ankiety zostaniemy przekierowani na podstronę wypełnionych ankiet.

Laravel

test

#	Ankieta
1	Ankieta do wypełnienia

hasło do odszyfrowania odpowiedzi

POKAŻ

Przypadek pokazuje, że nikt nie jest w stanie zobaczyć odpowiedzi użytkownika bez posiadania klucza, który został wcześniej podany przy wysyłaniu odpowiedzi. Aby użytkownik mógł zobaczyć wypełnioną przez siebie ankietę, musi podać klucz, w naszym wypadku kluczem jest słowo 'test'.

Po podaniu klucza oraz wciśnięciu przycisku 'POKAŻ' pojawiają się nam odpowiedzi, które zostały przez nas udzielone.

## Ankieta: Ankieta do wypełnienia

- Jak oceniasz, w skali od 1 do 6, gdzie: 1 - bardzo źle, 6 - bardzo dobrze, przygotowanie prowadzącego do zajęć? Zaznacz jedną odpowiedź.
  - 6
- Jak oceniasz, w skali od 1 do 6, gdzie: 1 - niezrozumiałe, 6 - zrozumiałe, jasność przekazywanych na zajęciach treści? Zaznacz jedną odpowiedź.
  - 6
- Płak oceniasz, od w skali od 1 do 6, gdzie: 1 - niepunktualny, 6 - punktualny, punktualność prowadzącego? Zaznacz jedną odpowiedź.
  - 6
- Jak oceniasz, od w skali od 1 do 6, gdzie: 1 - niewymagający, 6 - bardzo wymagający, poziom wymagań prowadzącego? Zaznacz jedną odpowiedź.
  - 6
- Jeżeli masz jakieś uwagi do zajęć lub prowadzącego, proszę wpisz je poniżej.
  - Testowy

Za odszyfrowanie danych odpowiada kod, który sprawdza czy konto jest kontem studenta, jeśli tak, sprawdza poprawność podanego klucza i na jego podstawie przechodzi do odszyfrowania odpowiedzi. Za cały ten proces odpowiada następująca część kodu

```
if (Auth::user()->is_student) {  
    $res = DB::select(DB::raw('value: "  
    SELECT  
        ua.answers as ans,  
        q.title  
    FROM user_answers ua  
    JOIN questionnaires q ON q.id = ua.questionnaire_id  
    WHERE ua.user_id = "' . $user->getAuthIdentifier() . "  
    AND q.id = "' . $request->input( key: "question_id")  
    ));  
    $data['questionnaire']['data'] = json_decode((new Crypton()->decryption($res[0]->ans,$request->input( key: "hash")), assoc: true);
```

Wykładowcy także mają możliwość podejrzenia wszystkich odpowiedzi jakie otrzymali od studentów. Warunkiem do tego koniecznym jest posiadanie konta, które nie jest kontem studenckim. Aby wiadomo było czy konto jest studenta czy wykładowcy w bazie danych mamy pole 'is\_student' które przyjmuje wartości 1 – dla studenta oraz 0 – dla wykładowcy.

is_student	tinyint(1)	<input type="text" value=""/>	<input type="text" value="0"/>
------------	------------	-------------------------------	--------------------------------

Po zalogowaniu się na konto wykładowcy otrzymujemy prosty widok który uprawnia nas tylko do sprawdzenia odpowiedzi jakie zostały udzielone przez studentów.

Ankiety

Po wejściu w zakładkę ‘Ankiety’ otrzymujemy widok ankiet które zostały oddane studentom do wypełniania.

#	Ankieta	
1	Ankieta do wypełnienia	<a href="#">POKAŻ</a>

Po wciśnięciu przycisku ‘POKAŻ’ zostajemy przekierowani do widoku odpowiedzi ankiety.

### Ankieta: Ankieta do wypełnienia

- Jak oceniasz, w skali od 1 do 6, gdzie: 1 - bardzo źle, 6 - bardzo dobrze, przygotowanie prowadzącego do zajęć? Zaznacz jedną odpowiedź.
  - 2,5,3,3,6,1,1,6,6,6
- Jak oceniasz, w skali od 1 do 6, gdzie: 1 - niezrozumiałe, 6 - zrozumiałe, jasność przekazywanych na zajęciach treści? Zaznacz jedną odpowiedź.
  - 4,4,4,4,6,3,2,5,6,6
- Płak oceniasz, od w skali od 1 do 6, gdzie: 1 - niepunktualny, 6 - punktualny, punktualność prowadzącego? Zaznacz jedną odpowiedź.
  - 6,5,3,3,6,5,3,5,6,6
- Jak oceniasz, od w skali od 1 do 6, gdzie: 1 - niewymagający, 6 - bardzo wymagający, poziom wymagań prowadzącego? Zaznacz jedną odpowiedź.
  - 2,4,4,4,6,2,4,6,6,6
- Jeżeli masz jakieś uwagi do zajęć lub prowadzącego, proszę wpisz je poniżej.
  - Text,asd,asd,asd,asd,asd,123,asdfasdf,Testowy,Testowy

Wszystkie odpowiedzi oddzielone są przecinkami, wykładowca widzi tylko odpowiedzi, nie ma możliwości w żaden sposób identyfikować osób z ich odpowiedziami.

Za powyższe wyświetlanie odpowiada kod

```

} else {
    $res = DB::select(DB::raw(
        value: "SELECT
        ua.answers as ans,
        q.title
        FROM questionnaire_answers ua
        JOIN questionnaires q ON q.id = ua.questionnaire_id
        WHERE ua.questionnaire_id = " . $request->input( key: "question_id")
    ));
    $answers = [];
    foreach ($res as $item){
        $itemArray = json_decode($item->ans, associ: true);
        foreach ($itemArray as $answer){
            if(isset( $answers[$answer["question"]])){
                $answers[$answer["question"]]["answer"] .= ",".$answer["answer"];
            }else{
                $answers[$answer["question"]] = $answer;
            }
        }
        // dd(json_decode($item->ans, true));
    }
    $data['questionnaire']['data'] = $answers;
    //dd($data);
}

$data['questionnaire']['ans'] = $res[0]->title;

```

Rozwiązanie, które zostało przez nas przyjęte daje możliwość szybkiego sprawdzania przez studentów czy udzielili odpowiedzi na ankietę jak również sprawdzenia odpowiedzi jakie zostały przez nich udzielone. Dodatkowym atutem naszego rozwiązania jest brak możliwości identyfikacji osób wypełniających ankietę zapewniając wszystkim całkowitą anonimowość. Unikalny klucz podawany przez każdą osobę wypełniającą ankietę daje pewność, że odpowiedzi nie mogą być w żaden sposób powiązane z ankietowanymi jednocześnie dając możliwość sprawdzenia odpowiedzi jakie zostały przez ankietowanego udzielone.