Rozdział 1

Dokumentacja

1.1 Połączenie z bazą danych

Kod realizujący połączenie z bazą danych, wykorzystywany na wszystkich stronach php.

```
1 $servername = "localhost";
2 $username = "root";
3 $password = "";
4 $dbname = "baza";
5 $connect = new mysqli($servername, $username, $password,
6                        $dbname);
```

Rys. 1.1: Laczenie z baza

servername - adres serwera
username - nazwa użytkownika
password - hasło
dbname - nazwa bazy danych
connect - otwiera nowe połączenie z serwerem MySQL.

1.2 Generowanie kodu hash

Kod pobiera dane z bazy MySQL, generuje kod hash oraz przesyła dane spowrotem do bazy. Jeżeli użytkownik udzielił odpowiedzi na ankietę aktualizuje pole w tabeli mail w celu zablokowania możliwości ponownego udzielenia odpowiedzi.

```
1  $wynik = $_POST['wynik'];
2  $a_id = "";
3  $o_id = "";
4  $user = $_SESSION['name'];
5  $hash = "";
6  $join = "";
7
8  for ($k = 1; $k < $wynik + 1; $k++) {
9      $odp = $_POST['q' . $k];
10     $que = $_POST['p' . $k];
11     $name = $_POST['nazwa_t'];
12
13     $sql = "SELECT * FROM odpowiedzi INNER JOIN pytania ON
14     odpowiedzi.o_p_id=pytania.p_id INNER JOIN ankieta ON
15     ankieta.a_id=pytania.p_a_id WHERE tworca='$user' AND
16     a_temat='$name' AND odpowiedz='$odp'";
17
18     $result = $connect->query($sql);
19     if ($result->num_rows > 0) {
20
21         while ($row = $result->fetch_assoc()) {
22             $a_id = $row['a_id'];
23             $o_id = $row['o_id'];
24
25         }
26     }
27
28     $sql1 = "INSERT INTO polacz(con_id, con_a_id, con_o_id)
29     VALUES ('','$a_id','$o_id')";
30     $result = $connect->query($sql1);
31
```

```
32      $join = $join . " " . $odp;
33
34  }
35  $sql2 = "UPDATE mail SET odpowiedz='1'
36          WHERE mail='$user'";
37  $result = $connect->query($sql2);
38
39  $join = $join . " " . $user . " " . $a_id;
40  $hash = md5($join);
41  $sql2 = "INSERT INTO hash(h_id, hash)
42          VALUES ('','$hash')";
43  $result = $connect->query($sql2);
```

Rys. 1.2: Przesylania hasha

wynik - pobranie wyniku za pomocą metody POST

a-id - id ankiety

o-id - id odpowiedzi

user - pobanie nazwy użytkownika za pomocą sesji

join - ciąg odpowiedzi, nazwy użytkownika oraz id ankiety

hash - ciąg join zamieniony na kod hash za pomocą metody md5

odp - odpowiedz przesana metodą POST

que - pytanie przesłane metodą POST

name - nazwa przesłana metodą POST

for(k = 1; k ¡ wynik + 1; k++)... - pętla odpowiada za połączenie ankiety z udzielonymi odpowiedziami przez użytkownika oraz wpisaniu odpowiedzi do łańcucha join

1.3 Zapytania do bazy danych

Kod odpowiada za realiowanie zapytań na bazie danych.

```
$sql = "SELECT * FROM odpowiedzi INNER JOIN pytania ON
odpowiedzi.o_p_id=pytania.p_id INNER JOIN ankieta
ON ankieta.a_id=pytania.p_a_id WHERE tworca='$user'
AND a_temat='$name' AND odpowiedz='$odp'";

    $result = $connect->query($sql);
    if ($result->num_rows > 0) {

        while ($row = $result->fetch_assoc()) {
            $a_id = $row['a_id'];
            $o_id = $row['o_id'];

        }
    }
$sql1 = "INSERT INTO polacz(con_id, con_a_id, con_o_id)
    VALUES ('','$a_id','$o_id')";
$result = $connect->query($sql1);
```

Rys. 1.3: Zapytania do bazy danych

sql - treść zapytania
result - rezultat dla zapytania
a-id - wartosc dla id ankiety
o-id - wartosc dla id odpowiedzi
if(result -num-rows 0) - sprawdza czy coś znajduje się w zmiennejresult
while (row = result fetch-assoc ( ))... - pętla przechodząca po tablicy asocjacyjnej

1.4 Sprawdzenie poprawności logowania

Kod odpowiada za logowanie do aplikacji.

```
$form_data = json_decode(file_get_contents("php://input"));

$validation_error = '';
```

```
4  //zabezpiecznie przed błędnym wprowadzeniem danych
5  if (empty($form_data->email)) {
6      $error[] = 'Email is Required';
7  } else {
8      if (!filter_var($form_data->email,
9              FILTER_VALIDATE_EMAIL)) {
10         $error[] = 'Invalid Email Format';
11     } else {
12         $data[':email'] = $form_data->email;
13     }
14 }
15
16 if (empty($form_data->password)) {
17     $error[] = 'Password is Required';
18 }
19 //jeśli dane są poprawne następuje zalogowanie
20 if (empty($error)) {
21     $query = "
22  SELECT * FROM users WHERE email = :email
23  ";
24     $statement = $connect->prepare($query);
25     if ($statement->execute($data)) {
26         $result = $statement->fetchAll();
27         if ($statement->rowCount() > 0) {
28             foreach ($result as $row) {
29                 if (password_verify($form_data->password,
30                                     $row["password"])) {
31                     $_SESSION["name"] = $row["email"];
32                 } else {
33                     $validation_error = 'Wrong Password';
34                 }
35             }
36         } else {
37             $validation_error = 'Wrong Email';
38         }
```

```
39      }
40  } else {
41      $validation_error = implode(", ", $error);
42  }
43
44  $output = array(
45      'error' => $validation_error,
46  );
47
48  echo json_encode($output);
```

Rys. 1.4: Logowanie

form-data - pobranie danych z inputa
if(empty(form-data-email)) - zabezpieczenie przed brakiem maila
if(empty(form-data-password)) - zabezpieczenie przed brakiem hasła
if (empty(error)) - jeśli nie ma błędów zaloguj w przeciwnym razie podaje komunikat błędu

1.5 Wykresy

Kod odpowiada za tworzenie wykresów z zebranych odpowiedzi.

```
1  <!DOCTYPE HTML>
2      <html>
3      <head>
4      <script>
5
6      window.onload = function() {
7      var chart = new CanvasJS.Chart("chartContainer", {
8        animationEnabled: true,
9        theme: "light2",
10       title:{
11         text: "Ankieta"
12       },
13       axisY: {
14         title: "Ilość odpowiedzi"
```

```
15        },
16        data: [{
17          type: "column",
18          yValueFormatString: "#,##0.## odpowiedzi",
19          dataPoints: <?php echo json_encode($tab,
20                              JSON_NUMERIC_CHECK); ?>
21        }]
22      });
23      chart.render();
24
25      }
26      </script>
27      </head>
28      <body>
29      <div id="chartContainer" style="height: 370px;
30                              width: 98%;"></div>
31      <script src="https://canvasjs.com/assets/script/
32                              canvasjs.min.js"></script>
33      </body>
34      </html>
```

Rys. 1.5: Wykres

1.6 Formularz pytań oraz odpowiedzi

Kod odpowiada za tworzenie pól potrzebnych do dodawania nowych pytań oraz odpowiedzi.

```
1 $(document).ready(function () {
2
3   $("#addq").click(function () {
4
5     smLib.surveyForms.addQuestion($("#questions"));
6   });
7 });
```

Rys. 1.6: dodawanie pytaĹ„

(”addq”).click(function () - Obsługuj użytkownika klikającego ”Add a question” poprzez dodanie nowego pytania
smLib.surveyForms.addQuestion((”questions”)) - zagnieżdżona funkcja tworzy strukturę HTML DOM.

Kod odpowiada za przekazywanie odpowiedzi oraz pytań z pliku JS do skryptu PHP

```
i = i - 1;
      var licznik = 'q' + i;
      var zlicz = 'radiochoice' + j;
      var z = j + 2;
      var resp = document.getElementById(licznik).value;
      var resp1 = document.getElementById(zlicz).value;
      $.post('php/insertquestion.php', {
        'pyt': resp
      });

      while (z < choiceX) {
        $.post('php/insertodpRadio.php', {
          'odp': resp1,
          'pyt': resp
        });
        j = j + 2;
        z = z + 2;
        zlicz = 'radiochoice' + j;
        resp1 = document.getElementById(zlicz).value;
      }
```

Rys. 1.7: przekazywanie z JS

document.getElementById().value; - bierze wartosc elementu na podstawie id
.post() ... - wysyla dane do okreslonego pliku w PHP

Funkcja odpowiada za tworzenie struktury pytań oraz odpowiedzi.

```
addQuestion: function (container) {...}
```

Rys. 1.8: struktuta pytan oraz odpowiedzi JS

Funkcja sprawdza dane użytkownika i w przypadku prawidlowych rozpoczyna sesję.

```
function register(buttonId)  {...}
```

Rys. 1.9: Logowanie JS

Bibliografia