



Politechnika Bydgoska
im. Jana i Jędrzeja Śniadeckich
Wydział Telekomunikacji, Informatyki i Elektrotechniki
Zakład Techniki Cyfrowej



Przedmiot:	Programowanie aplikacji sieciowych		Teleinformatyka Studia stacjonarne Semestr 5, 2021/2022
Temat:	System SCADA, IoT.		
Numer lab.:	4	Data wykonania:	2021.11.16
Prowadzący:	dr inż. Piotr Grad	Data oddania:	2021.11.16
Autor:	Anna Bagniewska	Indeks:	114881

1. Opis zadania:

Aplikacja ma zapewnić podgląd temperatur w pomieszczeniach. Z jednej strony mają zostać przesłane dane formularza i umieszczone w bazie danych. Następnie na witrynie mają zostać umieszczone elementy graficzne, tabele i wykresy, które mają ułatwić odbiorcy analizę sytuacji.

Sterownik (najczęściej poprzez sieć Internet) przekazuje dane z obserwowanego procesu do serwera na hostingu. Na serwerze wybrany skrypt PHP przekazuje dane do bazy MySQL. Następnie serwer przetwarza dane i wydaje odpowiednie polecenia innym częściom systemu.

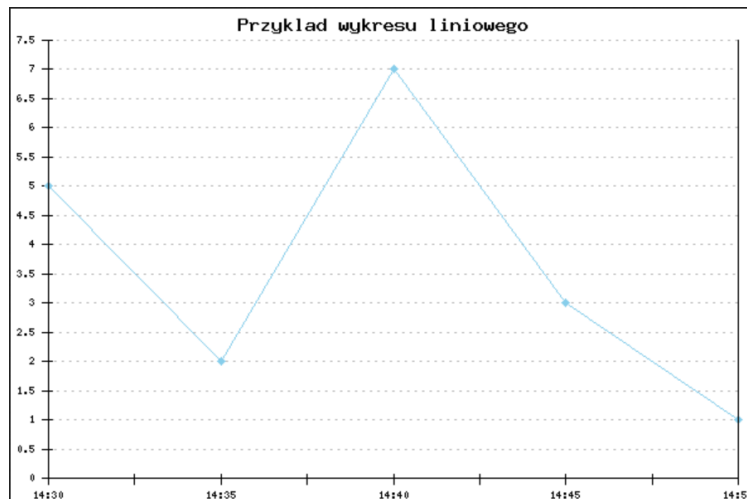
2. Opis programu:

I. Test działania pierwszego wykresu:







W folderze razem z plikiem skryptu wymagany jest plik `phplot.php`, na jego podstawie zostają wykreślane wykresy.

```
require_once 'phpplot.php';
```

Uzyskany wykres:



II. Utworzenie tabeli pomiaru:

					id	x1	x2	x3	x4	x5	datetime		
<input type="checkbox"/>		Edytuj		Kopijuj		Usuń	1	5	3	2	1	8	2021-10-25 09:45:43
<input type="checkbox"/>		Edytuj		Kopijuj		Usuń	2	3	2	-3	4	5	2021-11-03 17:34:13
<input type="checkbox"/>		Edytuj		Kopijuj		Usuń	3	2	3	3	2	1	2021-11-03 17:38:50

III. Utworzenie formularza:

Korzystając z możliwości wybrania typu wprowadzanych do formularza danych, ogranicza się wprowadzania np. liter i znaków specjalnych zamiast liczb.

```
x1:<input type="number"
```

x1: a

Aby zabezpieczyć formularz przed wysłaniem go z pustymi polami, należy skorzystać z parametru required.

```
x1:<input type="number" name="x1" min="-50" max="100" required><br>
```

x1: 4

x2:

x3:

x4:

x5:

Proszę wprowadzić liczbę

IV. Skrypt dodający dane do tabeli:

Zabezpieczenie skryptu przed typowymi błędami eksploatacyjnymi:

Brak wprowadzonych danych:

```
if (!$x1 || !$x2 || !$x3 || !$x4 || !$x5){
    echo "Nie wprowadzono danych do formularza";
}
```

Błąd połączenia z bazą danych:

```
$polaczenie = mysqli_connect($dbhost, $dbuser, $dbpassword, $dbname);
if (!$polaczenie) {
    echo "Błąd połączenia z MySQL." . PHP_EOL;
    echo "Errno: " . mysqli_connect_errno() . PHP_EOL;
    echo "Error: " . mysqli_connect_error() . PHP_EOL;
    exit;
}
```

Błąd zapytania MySQL:

```
or die ("Błąd zapytania do bazy: $dbname");
```

V. Tabela z danymi:

Bagniewska

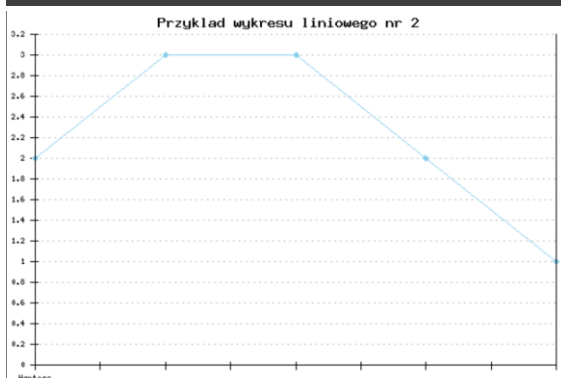
Panel Administracyjny - Bazy danych

https://abagniewska.beep.pl/z4/tabela.php

id	x1	x2	x3	x4	x5	Data/Godzina
1	5	3	2	1	8	2021-10-25 09:45:43
2	3	2	-3	4	5	2021-11-03 17:34:13
3	2	3	3	2	1	2021-11-03 17:38:50

VI. Wykres rysowany na podstawie ostatniego rekordu w bazie danych:

```
$rezultat = mysqli_query($polaczenie, "SELECT * FROM pomiary ORDER BY id DESC LIMIT 1");
```



VII. Wykreślanie kilku serii danych na wykresie:

Powyższe wykresy były generowane jako cały obrazek. W tej części wykres zostaje narysowany jako osobny element strony.

```

```

Kod odpowiedzialny za utworzenie kilku wykresów w jednym oknie składa się z kilku części: utworzenia nowych automatycznie numerujących się zmiennych:

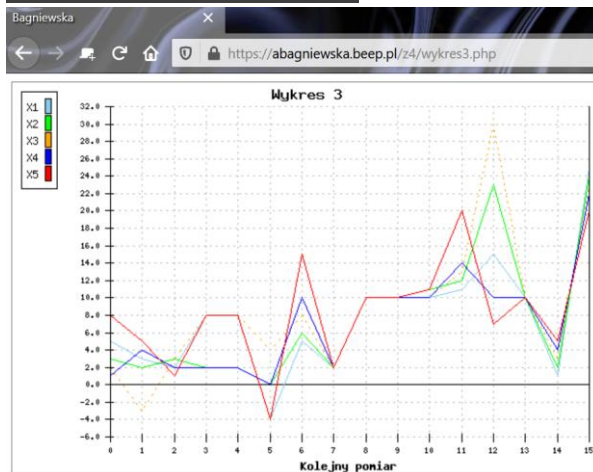
```
$k = 'k' . $ile;
```

przypisaniu im wartości zgodnie z kolejnymi iteracjami pętli while:

```
$$k = ' , ' . $ile . ' , ' . $x1 . ' , ' . $x2 . ' , ' . $x3 . ' , ' . $x4 . ' , ' . $x5;
```

przekształceniu ich na inny format, który jest obsługiwany przez phplot.php oraz wykreślenie samego wykresu:

```
array_push($dane, explode(' , ', $$k));  
$plot->SetDataValues($dane);
```



VIII. Wizualizacja systemów w budynku:

Wcześniejszy skrypt został rozbudowany o następujące polecenia:

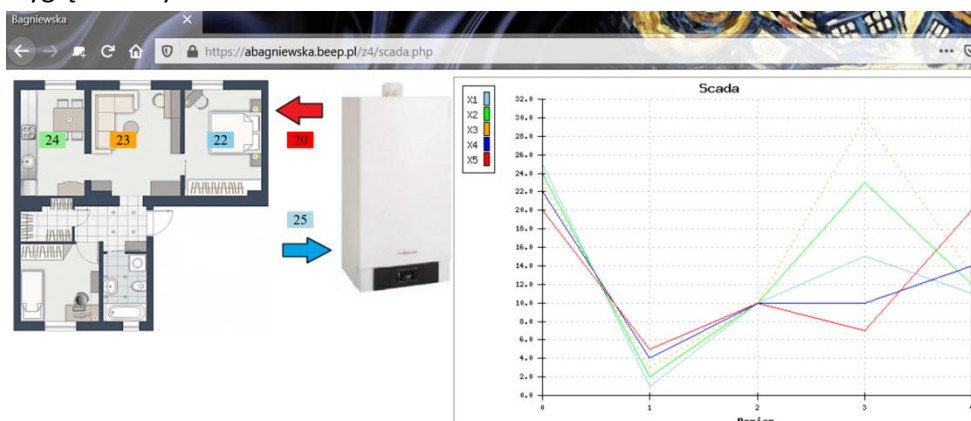
Odczytujące ostatnio wprowadzone dane:

```
while ($wiersz = mysqli_fetch_array($rezultat2)){  
    $id2 = $wiersz[0];  
    $wx1 = $wiersz[1];  
    $wx2 = $wiersz[2];  
    $wx3 = $wiersz[3];  
    $wx4 = $wiersz[4];  
    $wx5 = $wiersz[5];  
    $datetime = $wiersz[6];  
}
```

Umieszczające temperatury na planie budynku:

```
<div id="tablica" style="background: url('img/obraz3.png'); height: 291px; width: 500px;">  
  
<div id="zmienna" style="background-color: lightblue; text-align: center; float: left; width: 30px; position: absolute; top:  
160px; left: 320px;">  
<?php  
    echo $wx5;  
>  
</div>
```

Wygląd strony:



IX. Rozbudowa skryptu o inne czujniki:

Wymusiło to modyfikację formularza o przykładowy czujnik:

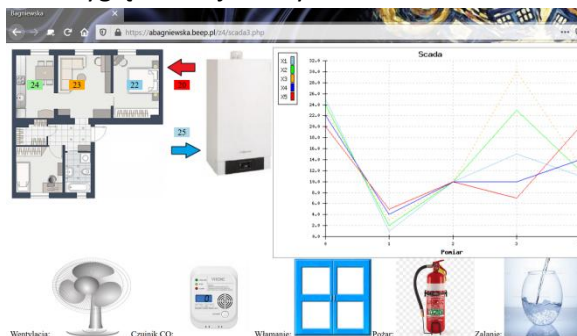
```
Pożar:
<select id="fire" name="fire">
  <option value=1>Tak</option>
  <option value=0 selected>Nie</option>
</select>
<br><br>
```

Oraz zmianę na stronie wskaźników:

```
<div id="gasnica" style="float: left;">Pożar:

</div>
```

Oraz wygląd samej strony:



X. Zadanie 10a

Realizacja tego zadania wymagała utworzenia nowej bazy danych, formularza oraz strony wizualizacji:

Baza danych:

id	nazwisko	obecność	obrazek
1	Polak	1	img/polak
2	Piasecki	1	img/piasecki
3	Nowak	1	img/nowak
4	Kowalski	1	img/kowalski
5	Lee	1	img/lee

Formularza imitującego system obecności:

```
<form method="POST" action="add10.php"><br>
  Kowalski:
  <select id="kowalski" name="kowalski">
    <option value=1>Obecny</option>
    <option value=0 selected>Nieobecny</option>
  </select>
<br><br>
```

Wraz ze skryptem obsługującym wprowadzone dane:

```
$r_kowalski = mysqli_query($polaczenie, "UPDATE pracownicy SET obecność='Kowalski' WHERE nazwisko='Kowalski'") or die ("Błąd zapytania do bazy air: $dbname");
```

Zakolorowanie nazwisk nieobecnych:

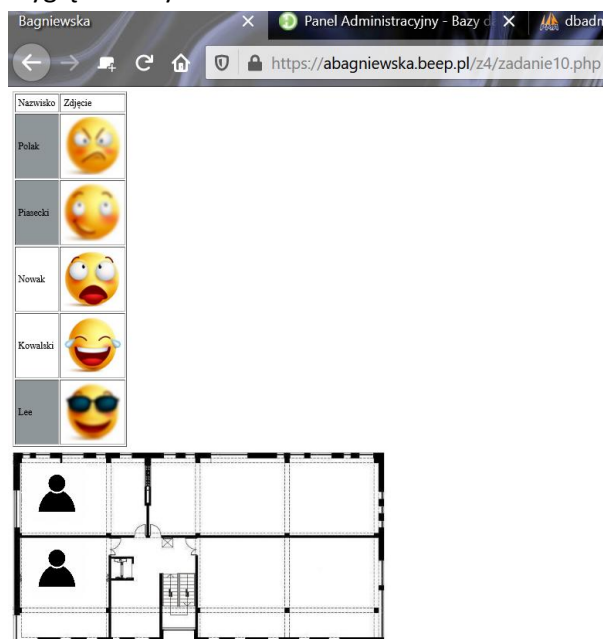
```
print "<tr><td style='background-color:$kolor;'>$nazwisko </td>
```

Wyświetlanie, które pomieszczenia budynku są zajęte realizuje po części funkcja

zliczania obecnych pracowników.

```
$zajete = mysqli_query($polaczenie, "SELECT SUM(obecność) FROM pracownicy") or die ("Błąd zapytania do bazy zajete: $dbname");
$pomieszczenie = mysqli_fetch_array($zajete);
if ($pomieszczenie[0]==1 | $pomieszczenie[0]==2 | $pomieszczenie[0]==3 | $pomieszczenie[0]==4 | $pomieszczenie[0]==5){
  echo 'img/osoba.gif';
}
```

Wygląd strony:



3. Wnioski:

W celu uniknięcia błędów w bazie danych, wszystkie wprowadzane przez użytkownika wartości są zabezpieczone i nie pozwalają pozostawić pustych pól – albo występuje monit o wprowadzenie poprawnych danych, albo zostają wprowadzone dane domyślne.