

SKI MANAGER

aplikacja korzystająca z bazy danych MySQL

Plan prezentacji

1. Funkcjonalność aplikacji
2. Omówienie kodu
3. MySQL
4. Wyrażenia regularne
5. Zadania



Do czego służy moja aplikacja?

-wybrane funkcjonalności

9:11

Ski Manager

Jan

Kowalski

1989.01.01

565762567

jankowalski@wp.pl

ZAREJESTRUJ SIĘ

ZALOGUJ SIĘ

registered successfully

9:12

Ski Manager

Email

Hasło

ZALOGUJ SIĘ

ZAREJESTRUJ SIĘ

REJESTRACJA I PANEL LOGOWANIA



9:16

Ski Manager

Wybierz swojego ulubionego instruktora

Grzegorz ▼

Wybierz dzień

< May 2020 >

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

ZATWIERDŹ

9:17

Ski Manager

Wybierz miejsce i godzinę

Ośla Łączka ▼

2 ▼

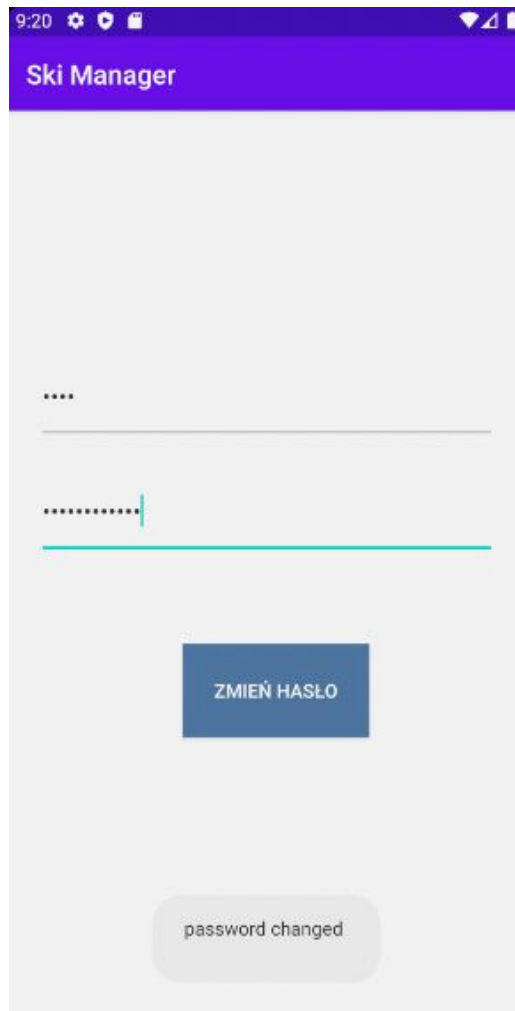
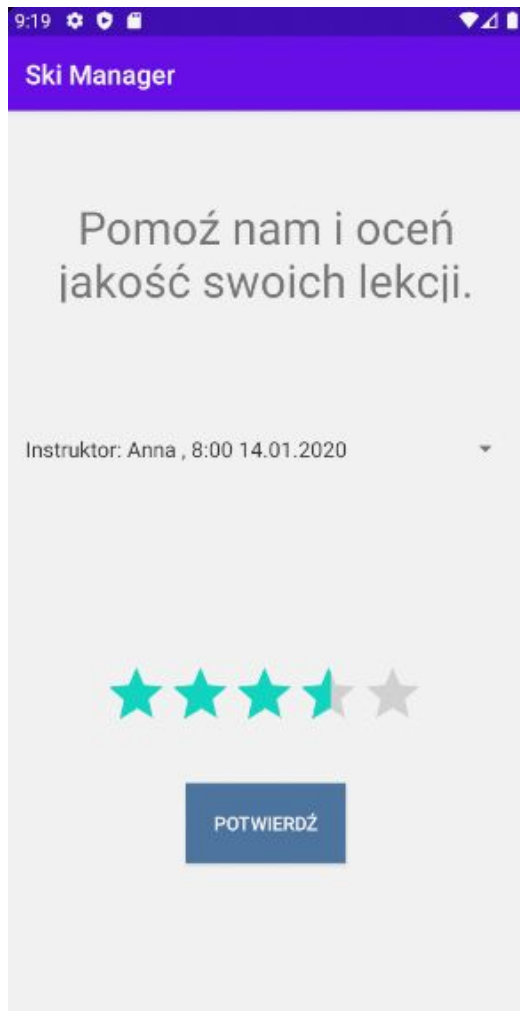
10:00 ▼

Ośla Łączka od 10:00 do 12:00

ZATWIERDŹ

lesson saved

ZAPISYWANIE LEKCJI



OCENA LEKCJI ORAZ ZMIANA HASŁA

Omówienie kodu

-połączenie z bazą danych

URLConnection

```
String loginURL="http://35.198.175.92/JPWP/login.php";

if(type.equals("login")) {
    String email = strings[1];
    String password = strings[2];
    try {
        URL url = new URL(loginURL);
        try {
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setDoOutput(true);
            httpURLConnection.setDoInput(true);
```


OutputStreamWriter i BufferedWriter

```
OutputStream outputStream = httpURLConnection.getOutputStream();
OutputStreamWriter outputStreamWriter = new OutputStreamWriter(outputStream, charsetName: "UTF-8");
BufferedWriter bufferedWriter = new BufferedWriter(outputStreamWriter);
String login_data = URLEncoder.encode( s: "email", enc: "UTF-8") + "=" + URLEncoder.encode(email, enc: "UTF-8") +
    "&" + URLEncoder.encode( s: "password", enc: "UTF-8") + "=" + URLEncoder.encode(password, enc: "UTF-8");
bufferedWriter.write(login_data);
bufferedWriter.flush();
bufferedWriter.close();

InputStream inputStream = httpURLConnection.getInputStream();
InputStreamReader inputStreamReader = new InputStreamReader(inputStream, charsetName: "ISO-8859-1");
BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
String result = "";
String line = "";
StringBuilder stringBuilder = new StringBuilder();
while ((line = bufferedReader.readLine()) != null) {
    stringBuilder.append(line).append("\n");
}
result = stringBuilder.toString();
```

PHP i MySQL - kod

```
<?php
$servername="localhost";
$username="root";
$password="123";
$dbname="ski";
$conn=mysqli_connect($servername, $username,$password, $dbname);
$email=$_POST['email'];
$rate='0';
$query="SELECT `instructor`, `year`, `month`, `day`, `hour` FROM `lessons` WHERE `email`='$email' AND `rate`='$rate'";
$result=mysqli_query($conn, $query);
if($result){
    while($row=mysqli_fetch_array($result)){
        $flag[]=$row;
    }
    print(json_encode($flag));
}
mysqli_close($conn);
?<img alt="lightbulb icon" data-bbox="38 865 55 895"/>
```

PHP i MySQL - funkcje

- `$conn = mysqli_connect($server, $user, $pass, $db)`
 - ▲ zwraca obiekt reprezentujący połączenie z bazą danych lub FALSE
- `$stmt = mysqli_prepare($conn, $query)`
- `mysqli_stmt_bind_param($stmt, $types, $var1, $var2...)`
 - ▲ jako \$types podajemy ciąg typów zmiennych, np. 'ids' (integer, double, string)
- `mysqli_stmt_execute($stmt)`
 - ▲ zwraca TRUE lub FALSE

PHP i MySQL - funkcje c.d.

- `mysqli_stmt_affected_rows ($stmt)`
 - ▲ stosujemy dla zapytań INSERT, UPDATE, lub DELETE, zwraca liczbę zmienionych wierszy
- `mysqli_stmt_bind_result ($stmt , $result1, $result2..)`
 - ▲ stosujemy dla zapytań typu SELECT, wyniki zapytania przechowujemy w zmiennych \$result1..
 - ▲ funkcja zwraca wartość TRUE lub FALSE
- `mysqli_stmt_fetch ($stmt)`
 - ▲ funkcja zwraca wartość TRUE lub FALSE

MySQL

-składnia zapytań

Zapytanie SELECT

SELECT * FROM registration

WHERE NOT (name = 'Anna' OR name = 'Janek')

ORDER BY surname

SELECT DISTINCT

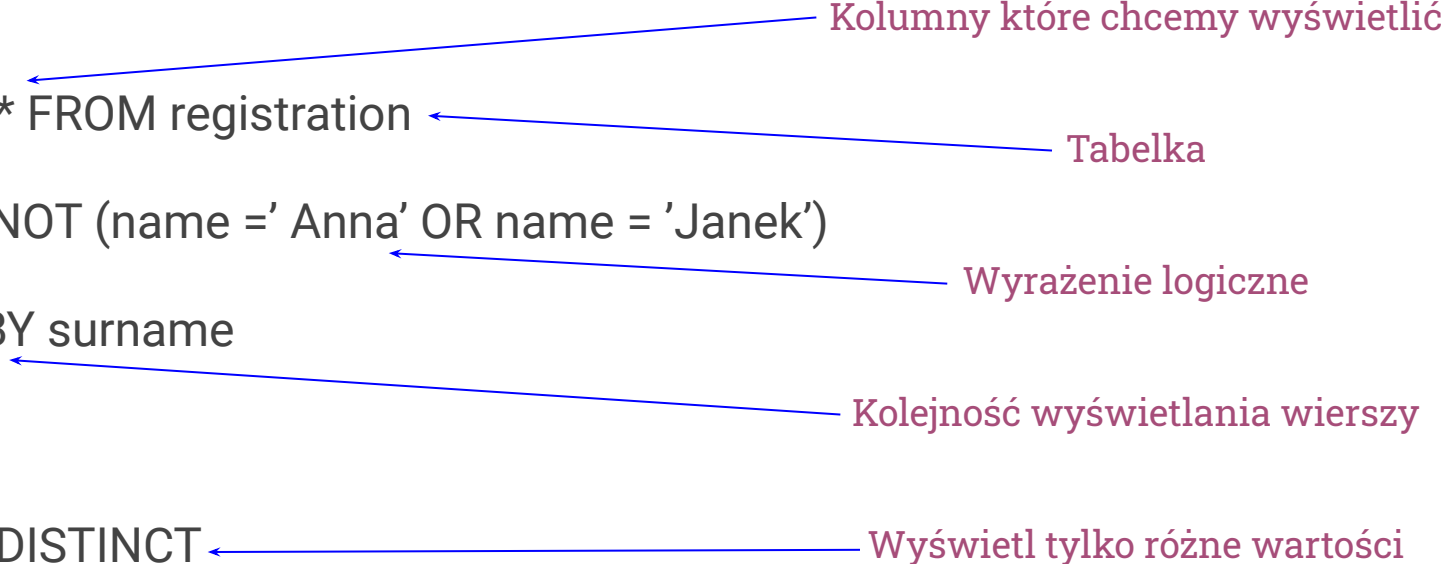
Kolumny które chcemy wyświetlić

Tabela

Wyrażenie logiczne

Kolejność wyświetlania wierszy

Wyświetl tylko różne wartości



The diagram illustrates the components of an SQL SELECT query. Blue arrows point from descriptive text on the right to specific parts of the query on the left. The query is split across five lines: 'SELECT * FROM registration', 'WHERE NOT (name = 'Anna' OR name = 'Janek')', 'ORDER BY surname', and 'SELECT DISTINCT'. The arrows connect the descriptions to the following parts: 'Kolumny które chcemy wyświetlić' to 'SELECT *', 'Tabela' to 'registration', 'Wyrażenie logiczne' to the 'WHERE' clause, 'Kolejność wyświetlania wierszy' to 'ORDER BY', and 'Wyświetl tylko różne wartości' to 'DISTINCT'.

Zapytanie INSERT INTO

```
INSERT INTO registration (name, surname) VALUES ('Jan', 'Kowalski')
```

Tabela

Nazwy Kolumn

Wartości



Zapytanie UPDATE

UPDATE registration ← Tabelka

SET surname = 'Malinowski' ← Wartości, które chcemy zmienić

WHERE name = 'Jan' ← Wyrażenie logiczne



Zapytanie DELETE

DELETE FROM registration WHERE name = 'Joanna' OR name = 'Katarzyna'

Tabela



Wyrażenie logiczne

REGEXP

czyli wyrażenia regularne

Wyrażenie regularne

- Wzorzec opisujący łańcuch symboli
- Pomaga sprawdzić zgodność ciągów tekstowych
- Duże ułatwienie dla programistów
- W większości języków programowania istnieją funkcje/biblioteki sprawdzające wyrażenia regularne
- W PHP wykorzystujemy `preg_match()`, która zwraca 1, 0 lub FALSE
- Wyrażenia przeważnie zapisujemy w `/.../`

Wyrażenia regularne

a-z Wszystkie Litery

0-9 Cyfry od 0 - 9

/d Wszystkie cyfry z różnych języków

/D Negacja /d

/s Puste pola

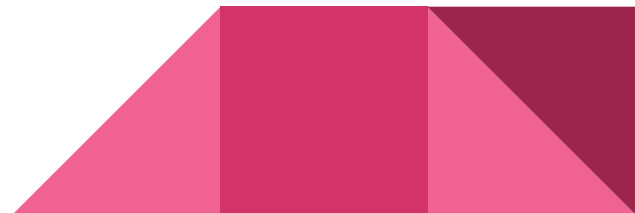
/S Negacja /s

/w Wszystkie słowa

/W Negacja /w

. Dowolny znak

/+ /. Znak '+' oraz znak '.'



Wyrażenia regularne - Flagi

- `/ ... /g` Przeszukuje globalnie, zwraca więcej niż jedną zgodność (**g**lobal)
- `/ ... /i` Nie zwraca uwagi na wielkość liter (case **i**nsensitive)
- `/ ... /m` Gdy użyjemy znaków `^` `$` sprawdza dla każdej linii osobno (**m**ultiline)
- `/ ... /s` Jeżeli użyjemy `.` zgodność wszędzie oprócz nowej linii (**s**ingleline)
- `/ ... /y` Gdy znajdzie pierwsze dopasowanie nie przeszukuje dalej (stick**y**)



/abc/ Dokładne odwzorowanie abc

/ie/g

Text Tests NEW

To jest pierwsze zdanie.

/abc+/ Zgodność dla ab i 1 lub więcej c obok siebie

/an+/g

Text Tests NEW

To jest staranne zdanie.

/a(bc)?/ Zgodność dla a i 0 lub 1 ciągu bc

/an?/g

Text Tests NEW

To jest staranne zdanie.

/a*bc/ Zgodność dla 0 lub więcej a i bc

/an*/g

Text Tests **NEW**

To jest staranne zdanie.

/ab{3}/ Zgodność dla a i b dokładnie 3 razy

/an{2}/g

Text Tests **NEW**

To jest staranne zdanie.

/a(bc){,4}/ Zgodność dla a i bc maksymalnie 4 razy

/(an){2,}/g

Text Tests **NEW**

To jest staranne zdanie.

`/[abc0-9]/`

Zgodność dla a lub b lub c lub cyfry



`/(aa|AA)/`

Zgodność dla aa lub AA



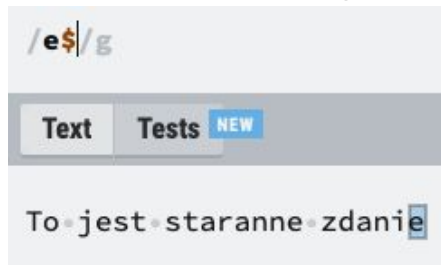
`/^(Aa)/`

Zgodność dla ciągów zaczynających się od Aa



/c\$/

Zgodność dla ciągów kończących się literą c



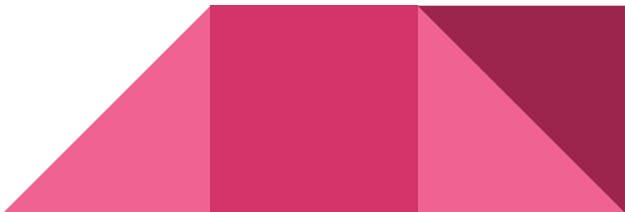
/a(?=b)/

Zgodność dla a, po których znajduje się b



/(?<=a)b/

Zgodność dla b, które poprzedza a



Ważny przykład

```
/[a-zA-Z]{1,}/g
```

Text Tests **NEW**

1234Martyna1234

```
/^[a-zA-Z]{1,}$/g
```

Text Tests **NEW**

1234Martyna1234

Zadania

<https://github.com/martynapawlus/JPWP>

Pomocne linki

- https://developer.android.com/preview?gclid=CjwKCAjwh472BRAGEIwAvHVfGkaGt2wjPi4_z9Xq3LFs8UJiK7OOZ2rZd8x0Bn9-UOxgHgQl0Uk4tRoC5AgQAvD_BwE&gclsrc=aw.ds
- <https://www.php.net/manual/en/index.php>
- <https://www.w3schools.com/sql/>
- <https://www.youtube.com/watch?v=rhzKDrUiJVk>
- <https://www.youtube.com/watch?v=yUWuQTQw7tc>
- <https://docs.oracle.com/javase/8/docs/api/java/net/URLConnection.html>

