KIV / WEB

Webové aplikace

7. cvičení

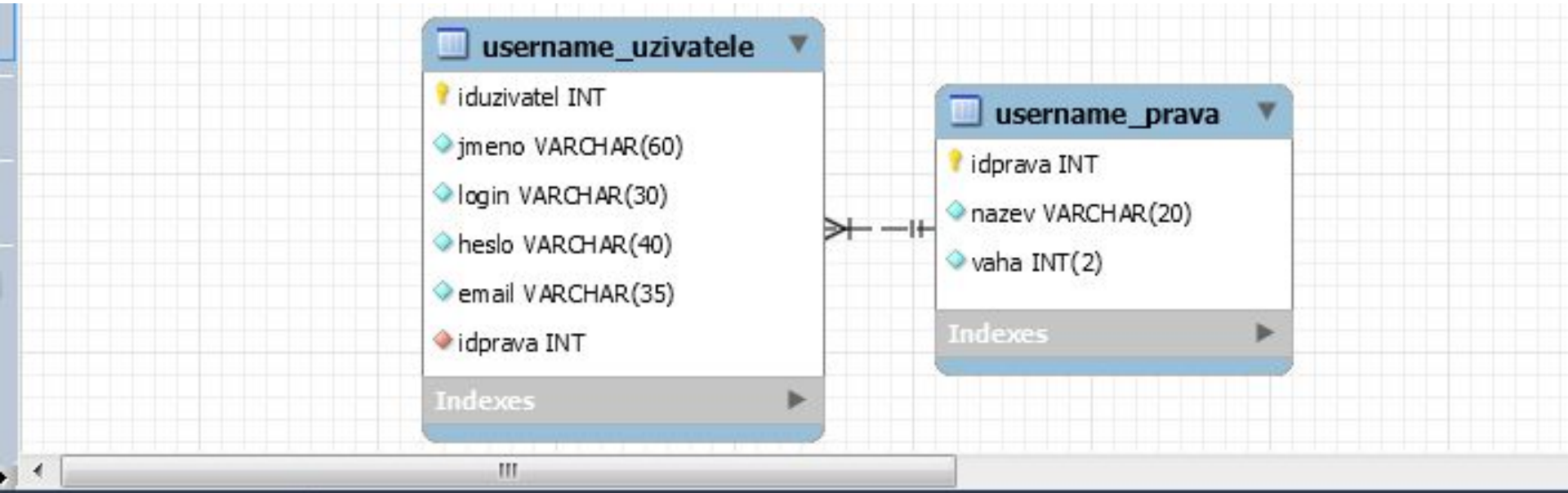
zimní semestr 2015/16

Cvičící: Michal Nykl

KIV server pro vývoj web. aplikací

- students.kiv.zcu.cz
  - přihlášení: orion login a heslo
- Složka:
  /afs/kiv.zcu.cz/kiv/home/student/***orion_login***/
  /public-kiv/public_html
- Zobrazení:
  students.kiv.zcu.cz/~orion_login/

- Soubory pro 7. cvičení



name_uzivatele - Table

Table Name: username_uzivatele Schema: **mydb**

Filter Rows: Edit: Export/Import: Wrap Cell Content: Apply changes:

iduzivatel	jmeno	login	heslo	email	idpr
1	Pokusný administrátor	admin	admin	pokus1@kiv.zcu.cz	1
2	Pokusný recenzent	pokus	pokus	pokus2@kiv.zcu.cz	2
NULL	NULL	NULL	NULL	NULL	NULL

- nahraďte v SQL skritpu “username” za orion login
- vytvořte tabulky v databázi a zprovozněte soubory
  - upravte jméno tabulky v databaze.class.php

Práce s databází: PDO

- new PDO("mysql:host=$host;dbname=$dbname",'log','pas')
- položení dotazu - $dotaz = $db->query($dotaz)
- čtení výsledků po řádcích:

```
while($row = $dotaz->fetch(PDO::FETCH_ASSOC)){
    $pole[] = $row['login'].'<br>'; // název sloupce tab.
}
```

- uložení všech výsledků do pole:

```
$pole = $dotaz->fetchAll();
```

Práce s databází: PDO a dotazy

- klasický dotaz pro výběr:
- $dotaz = $db->query('SELECT * FROM users');
-
- dotaz SELECT (v krátkosti):
- SELECT *

  FROM uzivatele, prava

  WHERE login=’admin’ AND heslo=’pas’

  AND uzivatele.idprava = prava.idprava;

  (cizí klíč) (primární klíč)

Útok: SQL injection

- máme dotaz pro přihlášení uživatele:
  $dotaz = "SELECT * FROM nyklm_uzivatele WHERE login='$log' AND heslo='$pas';"

- do hesla v HTML inputu vložíme: **' OR '1'='1**
- tj. vznikne:

  "…. AND heslo='$pas' **OR '1'='1**';"

- které je vždy splněné, tj. získáme všechny uživatele

PDO obrana proti SQL injection

- využitím předpřipravených dotazů:

```
$sql = "SELECT * FROM nyklm_uzivatele
                WHERE login=:log AND heslo=:pas;"

$params = array(':log' => $log, ':pas' => $pas);
$dotaz= $db->prepare($sql);

// provede dotaz
if(!$dotaz->execute($params)){
        return null; // dotaz nemá vysledek
}

// čte výsledky
$pole = $dotaz->fetchAll();
```

Práce s databází: PDO a dotazy

- INSERT INTO
  nyklm_uzivatele (jmeno, login, heslo, email)
  VALUES ('jméno', 'log', 'pas', 'mail@kiv.zcu.cz'),
  ('jméno2', 'log2', 'pas2', 'mail2@kiv.zcu.cz');

- iduzivatele je Auto Increment, idprava je defaultně 3
- následně lze volat: $db->lastInsertId();

- lze použít předpřivaný dotaz, např. pouze ve tvaru:

```
$sql = "INSERT INTO
        nyklm_uzivatele (jmeno, login, heslo, email)
    VALUES (?,?,?,?)";
$dotaz = $db->prepare($sql);
$dotaz->execute(array($jm, $log, $pas, $mail));
```

Úkoly

- úkol: ošetřete SQL Injection u SELECT dotazu
- úkol: doplňte funkce pro registraci uživatele

Útok: Cross-site scripting (XSS)

- vložení “útočného” HTML či JavaScriptu do stránky
- např.:

```
<p style="position:fixed;top:0px;left:0px;">
    ÚTOČNÝ KÓD
</p>
```

Cross-site scripting (XSS)

- na serveru nutné escapování znaků,
  tj. převod < na < a > na > apod.
- funkce:

  $text = **htmlspecialchars**($text);

- a vynucení číselných hodnot:

  $cislo = $_POST[“cislo”] **+ 0**; // =0

- neuchrání uživatele před podstrčeným XSS, který se netýká serveru, např. jako součást GET požadavku v odkazu (zde ale opět chyba serveru):
- ?vstup=ahoj<script>window.alert("ÚTOK");</script>

Práce s databází: PDO a dotazy

- **Mazání:**
  - **DELETE FROM uzivatele**
    WHERE iduzivatele=112;

- **Úprava:**
  - **UPDATE uzivatele**
    SET jmeno='asdf', email='asd@asd.cz'
    WHERE iduzivatele=2;

- Opět lepší využít předpřipravené dotazy

Úkoly

- ošetřete XSS v dotazech
- doplňte stránku uživatele o možnost smazání účtu
- doplňte stránku o editaci osobních údajů uživatele

Děkuji vám za pozornost.

příště: pokračování PHP + MVC