

Računarski praktikum 2 – Druga zadaća: Quack

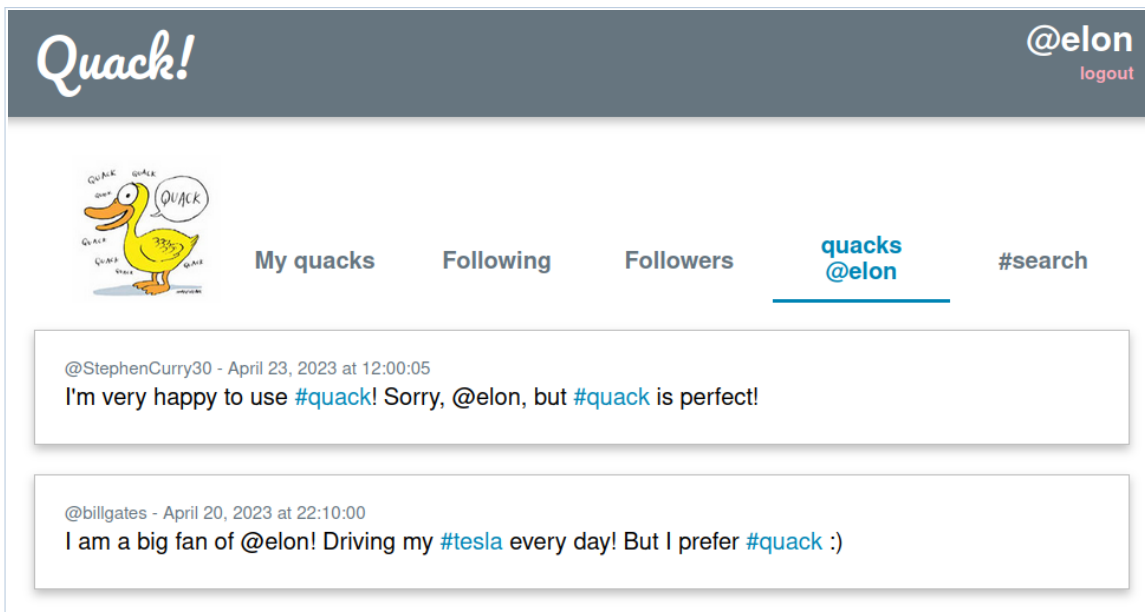
Mirko je došao na originalnu ideju za web-aplikaciju imena *Quack*. U toj aplikaciji:

- Korisnici mogu objavljivati kratke poruke (tzv. *quack*), duljine max. 140 znakova.
- Poruke mogu sadržavati oznake (tzv. *tag*), koje imaju oblik #oznaka.
- Poruke mogu sadržavati korisnička imena, koja imaju oblik @username.
- Korisnici mogu pratiti (*follow*) jedni druge.

Pomozite Mirku da implementira ovu web-aplikaciju u PHP-u.

Konkretno, potrebno je implementirati sljedeće:

- Na adresi <https://web.math.pmf.unizg.hr/nastava/rp2d/slideovi/2023/prepareDB.php.txt> možete skinuti datoteku `prepareDB.php` koja inicijalizira tri tablice u bazi podataka.
 1. `dz2_users` sadrži podatke o korisnicima *Quacka* (id, korisničko ime, hashirani password, e-mail, niz za registraciju, te oznaku o uspješnoj registraciji).
 2. `dz2_follows` sadrži podatke o tome koji korisnik prati kojeg (id osobe, id osobe koju slijedi).
 3. `dz2_quacks` sadrži podatke o porukama (id poruke, id usera koji ju je objavio, sadržaj poruke, vrijeme objavljivanja).
- Prilagodite podatke za spajanje na bazu, te inicijalizirajte tablice unutar svoje baze na rp2 serveru tako da izvršite skriptu `prepareDB.php`.
- Kada korisnik pristupi početnoj stranici aplikacije, prikazuje mu se forma za ulogiravanje.
- Po uspješnom ulogiravanju (provjera username / password), korisniku se prikazuje izbornik u kojem može odabrati prikaz:
 1. Svih *quackova* koje je on dosad objavio. Tu može napisati/objaviti novi quack (sami odaberite način unosa; dovoljan je obični textbox).
 2. Svih *quackova* koje su objavili drugi korisnici koje on prati. Tu može dodati ime novog korisnika kojeg će pratiti ili ukloniti nekog kojeg već prati (sami odaberite način unosa; dovoljan je obični textbox).
 3. Imena svih korisnika koji ga prate.
 4. Svih *quackova* u kojima se pojavljuje njegov username (npr. @Mirko).
 5. Forme u koju može unijeti neki tag (npr. #croatia), te onda prikazati sve *quackove* u kojima se pojavljuje taj tag.
- Nakon ulogiravanja se (osim izbornika) prikazuje ono što je opisano u točki 1.
- *Quackove* uvijek treba prikazivati silazno sortirano po datumu objave (dakle, prvo najnovije).
- Treba omogućiti i odlogiravanje korisnika.
- Bonus za dodatni bod: Kad god se u nekom *quacku* pojavljuje hashtag, taj hashtag treba prikazati kao link koji vodi na prikaz iz točke 5.
- Primjer prikaza stranice iz točke 4 (ne morate napraviti identičan; izgled nije bitan nego gore opisana funkcionalnost):



Bonus za još 2 dodatna boda (povrh onog jednog gore):

- Uočite da gore opisani zadatak ne zahtijeva dodavanje novih korisnika. U tablici `dz2_users` vam zapravo trebaju samo prva tri stupca (`id`, `username`, `password_hash`). Za dodatna 2 boda, omogućite da korisnici otvaraju nove korisničke račune na način na koji je to opisano u Zadatku 6 iz Zadataka za vježbu. Za to će vam trebati i preostala tri stupca u `dz2_users`.

Bonus za još 1 dodatni bod:

- Zadatak implementirajte korištenjem objektno-relacijskog preslikavanja (ORM), kako je opisano u Predavanjima 7. ORM trebate implementirati tako da su klase u modelu izvedene iz bazne klase `Model`. U baznoj klasi se ne smijete referirati na pojedine izvedene klase (tj. imati nešto poput `if(imelzvedeneKlase === 'User') ...`), nego ona treba biti posve generička. Smijete se referirati na popis i tipove stupaca (vidi iduću točku).
- U bazoj klasi `Model` smijete uz `$table` dodati i popis atributa `$attributes` kao statičko polje. U tom polju će izvedena klasa navesti popis svih stupaca u pripadnoj tablici i njihovih tipova, npr.

```
class User extends Model {  
    protected static $table = 'dz2_users';  
    protected static $attributes = ['id'=>'int', 'name'=>'string', 'email'=>'string', ...];  
}
```

Sugestije:

- Uočite da je praktički jedina razlika kod točaka 1-5 u SQL upitu i da je ispis *quackova* dovoljno implementirati samo jednom (kao jednu funkciju ili view) za sve podstranice.
- Jako se preporuča se korištenje MVC (jer olakšava organizaciju koda), ali ono nije nužno.
- Svakako koristite *prepared statements* za unos *quackova* u bazu!
- Za pristup *quackovima* koji sadrže na primjer hashtag jednak `#Lakers` možete koristiti link oblika: `quack.php?hashtag=Lakers`, pa onda u toj skripti dohvatiti `$_GET['hashtag']`.
- Vezu između tipova podataka za spremanje vremena u MySQL i PHP možete vidjeti npr. ovdje: <https://www.pixelstech.net/article/1318874457-Dates-in-PHP-and-MySQL>

Upute za postavljanje aplikacije na web i slanje rješenja:

- Aplikacija se smije sastojati od jedne ili više PHP, HTML, CSS i slikovnih datoteka.
- Za rješavanje nemojte koristiti JavaScript, nego isključivo PHP.

- Sve datoteke od kojih se sastoji aplikacija zapakirajte u datoteku `quack.zip` (čak i u slučaju kad se sastoji samo od jedne datoteke!), te tu datoteku uploadajte kroz sustav za zadaće na adresi: <https://web.math.pmf.unizg.hr/nastava/rp2d/zadace.php>
- Nadalje, aplikaciju postavite tako da bude dostupna preko rp2-servera kroz ovu adresu (pazite na velika i mala slova!):
<https://rp2.studenti.math.hr/~username/dz2/quack.php>
Prava pristupa za direktorij `/student1/username/public_html/rp2/dz2` prije početka izrade zadaće postavite na `rwX-----X` (701).
- Koristite svoju MySQL bazu na rp2 serveru.