

Technická dokumentace

Stack

Webová stránka je postavená v HTML, CSS a JS na front-endu. Na backend byl použit jazyk PHP. Jako web server je používán Apache. Data, se kterými stránka pracuje jsou uložena v JSON souboru.

Schéma sessions

Při úspěšném přihlášení dojde k uložení username do superglobální proměnné `$_SESSIONS` pod klíčem „username“. Je to používáno k ověření, zda je uživatel přihlášen a k získání přihlášeného uživatele z JSON souboru.

Schéma JSON souborů

users.json

```
{"id":"675c513a6e709",  
"email":"admin@seznam.cz",  
"username":"FilipAdmin",  
"password":"hashedpasswordblabla",  
"role":"admin"}
```

Všechny údaje kromě password a role jsou pro každého uživatele unikátní. Unikátnost jmen je ověřována pomocí AJAXu v ajax.js a users_ajax.php.

inzeraty.json

```
{"id":"675c5178c7b0d",  
"lokalita":"Praha",  
"cena":"40 000",  
"mena":"czk","rozmary":"40",  
"popis":"utulny bytecek",  
"prodej":"pronajem",  
"img":{"name":"someblaid.jpg","full_path":"soubory/img","type":"image/jpeg","tmp_name":"\\tmp\\phpU2Rlt7","error":0,"size":6615},  
"user_id":"675c517778c95"}
```

Každý inzerát má své unikátní id, které je předáváno přes hidden formulářové pole a id uživatele, který ho přidal. Další klíče obsahují dodatečné informace.

Co mají jednotlivé soubory na starosti?

[signup.php](#)

Účel: Registrovat uživatele

Uživatel zadává do formulářových polí unikátní údaje, kromě hesla. Pokud dojde k odeslání formuláře, všechny políčka jsou validována. Pokud dojde k chybě, formulář se vrátí předvyplněný a sdělí uživateli co je špatně. K předvyplnění dojde přes superglobální proměnnou POST, kdy se formulář posílá sám na sebe. Jestliže dojde ke správnému vyplnění údajů, uživatel je přidán do users.json s hashovaným heslem, unikátním id, které bylo posláno přes hidden formulářové pole a všemi vyplněnými údaji. Také se nastaví superglobální proměnná SESSION, ve které je klíč „username“. Poté dojde k přesměrování na index.php. Pomocí JS je realizována kontrola obsazenosti username a validace hesel.

[login.php](#)

Účel: Přihlašovat uživatele

Uživatel vyplní formulářová pole, pokud dojde k odeslání, opět se validuje vstup. Pokud není něco v pořádku, vrátíme uživatele na předvyplněný formulář (opět pomocí POST) s chybovou hláškou. Pokud jsou data validní, dojde k přesměrování na index.php a nastavení SESSION proměnné na username uživatele pod klíčem „username“.

[functions.php](#)

Účel: Extrahovat většinu opakujících se funkcí do jednoho souboru

Většina funkcí, co obsluhuje webovou aplikaci.

[index.php](#)

Účel: Zobrazit inzeráty z JSON souboru

Na stránce se zobrazují všechny inzeráty z inzeraty.json, které jsou odkazem na podrobnější inzerát. Anchor element nastaví cílovou adresu na inzerat.php?id= id inzerátu. Všechny jsou seřazeny vzestupně podle rozměrů. Jsou vypisovány pomocí foreach loopu, kdy procházíme asociativní pole z json souboru. Stránkování je realizováno pomocí nastavení limitu na každou stránku a následně tím obohatí URL, určíme si aktuální stránku, také obohatíme URL, a poté už jen vypočítání, odkud budeme inzeráty brát na aktuální stránku (na stránce 3 (offset = (3-1)*náš limit na stránku)).

[inzerat.php](#)

Účel: Zobrazit podrobnosti o inzerátu

Na stránku je možno se dostat rozkliknutím inzerátu na index.php. Inzerát se zobrazuje podle id, které bylo získáno ze superglobální proměnné GET. Podle id se pak najde

pomocí foreach objekt inzerátu v jsonu, ze kterého se následně vypisují data. Pokud je uživatel adminem, může příspěvek smazat. Pokud je uživatel zároveň tím, který inzerát přidal, může inzerát upravovat. Pokud uživatel není ani jedno, tak se mu nezobrazí ani jedna možnost. Toto je zjišťováno pomocí SESSION superglobální proměnné, kde se nejdříve získá pomocí klíče „username“ objekt uživatele z users.json a následně přes klíče přistupujeme k roli a id. Pokud se uživatelovo id shoduje s id uživatele, co příspěvek přidal, tak se zobrazí co je zmíněno výše. To samé s adminem. Pokud uživatel klikne na smazat, přesměruje ho to na deleteAd.php s obohacením odkazu o id inzerátu. Pokud uživatel klikne na upravit, přesměruje ho to na upravit.php s obohacením odkazu o id inzerátu.

[upravit.php](#)

Účel: Upravovat inzeráty

Na tuto stránku se lze dostat přes odkaz v detailu inzerátu. Uživatel musí být přihlášen a musí být inzerát jeho, aby mohl upravovat. Toto je kontrolováno pomocí SESSION superglobální proměnné. Poté se z GET proměnné získá id inzerátu, které se využije k identifikaci objektu v inzeraty.json. Pokud neexistuje, je přesměrován na index. Pokud existuje, tak dojde k předvyplnění dat. Po odeslání dojde k jejich validaci, pokud je něco špatně, uživatel je upozorněn. Pokud je vše validní, dojde k přepsání v json souboru a následnému přesměrování na upravený inzerát.

[addform.php](#)

Účel: Přidávat inzeráty

Uživatel musí být přihlášen, jinak je přesměrován na přihlášení. Uživatel vloží data, která jsou validována. Pokud není něco v pořádku, uživatel je upozorněn a dojde k vrácení formuláře s předvyplněnými daty přes POST superglobální proměnnou. Pokud je vše validní, dojde k uložení do inzeraty.json. Každý inzerát má své unikátní id a přes hidden formulářové pole dochází k ukládání id uživatele, který inzerát přidal. Obrázek se uloží přeformátovaný, aby nedocházelo ke zbytečnému načítání. Následně je uživatel přesměrován na index.php.

[profil.php](#)

Účel: Zobrazení uživatelského profilu

Uživatel se ke stránce opět může dostat pouze jako přihlášený, poté se najde uživatel v users.json pomocí klíče „username“ v superglobální proměnné SESSIONS a vypíše se údaje z JSON souboru – username, email, role.

[header.php](#)

Účel: Start session, načíst uživatele

Pomocí SESSION superglobální proměnné načte uživatele z json souboru, pokud je nastavena, jinak vrátí null.

logout.php

Účel: Odhlásit uživatele

Vymaže obsah celé superglobální proměnné SESSION, tím dojde k odhlášení uživatele, a přesměrování na login.php.

vypisuzivatelu.php

Účel: Správa rolí pro adminy

Pokud uživatel není přihlášen nebo není admin, dojde k přesměrování na login. Pokud je admin, dojde k vypsání uživatelů pomocí foreach procházení users.json. U každého je select aktuální role. Pokud si admin přeje roli změnit, stačí vybrat roli a potvrdit submittem. Poté dojde přes POST proměnnou k přepsání dat v users.json a přesměrování na index.php.

nav.php

Účel: Template pro navigační menu

user_ajax.php

Účel: Kontrola unikátnosti username

Načte uživatele z users.json, následně vytvoří pole s uživatelskými jmény. Poté ajax.js pošle XMLHttpRequest, ve kterém je hodnota username a user_ajax.php vrátí buď string used – obsazeno nebo string not_used – neobsazeno. Podle toho pak vypíše signup.php chybovou hlášku, či nikoliv.

deleteAd.php

Účel: Smazat inzerát

Po kliknutí na tlačítko smazat na daném inzerátu dojde k získání jeho id přes GET superglobální proměnnou, poté prohledá všechny inzeráty v inzeraty.json. Pokud se id inzerátu neshoduje, je přidáno do pole zbývajících inzerátů. Toto pole následně nahradí obsah JSON souboru – ale bez inzerátu s daným id z GET proměnné. Dojde k přesměrování na index.php.

