Administrátorská dokumentace

Logos polytechnikos

Dokumentace určená ke správě školního internetového časopisu Vysoké školy polytechnické v Jihlavě s názvem Logos polytechnikos od týmu Tarzani

[Vysoká škola polytechnická Jihlava]

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Telefon** 567 141 111  **E-mail** vspj@vspj.cz | Tolstého 1556 586 01 Jihlava | https://alpha.kts.vspj.cz/~pribyl32/rsp/  Team Tarzani |
|  |  |  |

Obsah

[Instalace programů pro vývojové prostředí 1](#_Toc60316851)

[Přehled potřebných programů 1](#_Toc60316852)

[Instalace XAMPP s PHP 2](#_Toc60316853)

[Instalace CA certifikátu 3](#_Toc60316854)

[Instalace Composeru pro PHP 4](#_Toc60316855)

[Instalace Symfony 6](#_Toc60316856)

[Instalace WinSCP 7](#_Toc60316857)

[Instalace aplikace GitHub 8](#_Toc60316858)

[Spuštění webu na školním serveru 9](#_Toc60316859)

[Konfigurace uživatelských rolí 12](#_Toc60316860)

[Konfigurace implementace databáze 13](#_Toc60316861)

[Správa na webové stránce 15](#_Toc60316862)

[Podpora 19](#_Toc60316863)

Instalace programů pro vývojové prostředí

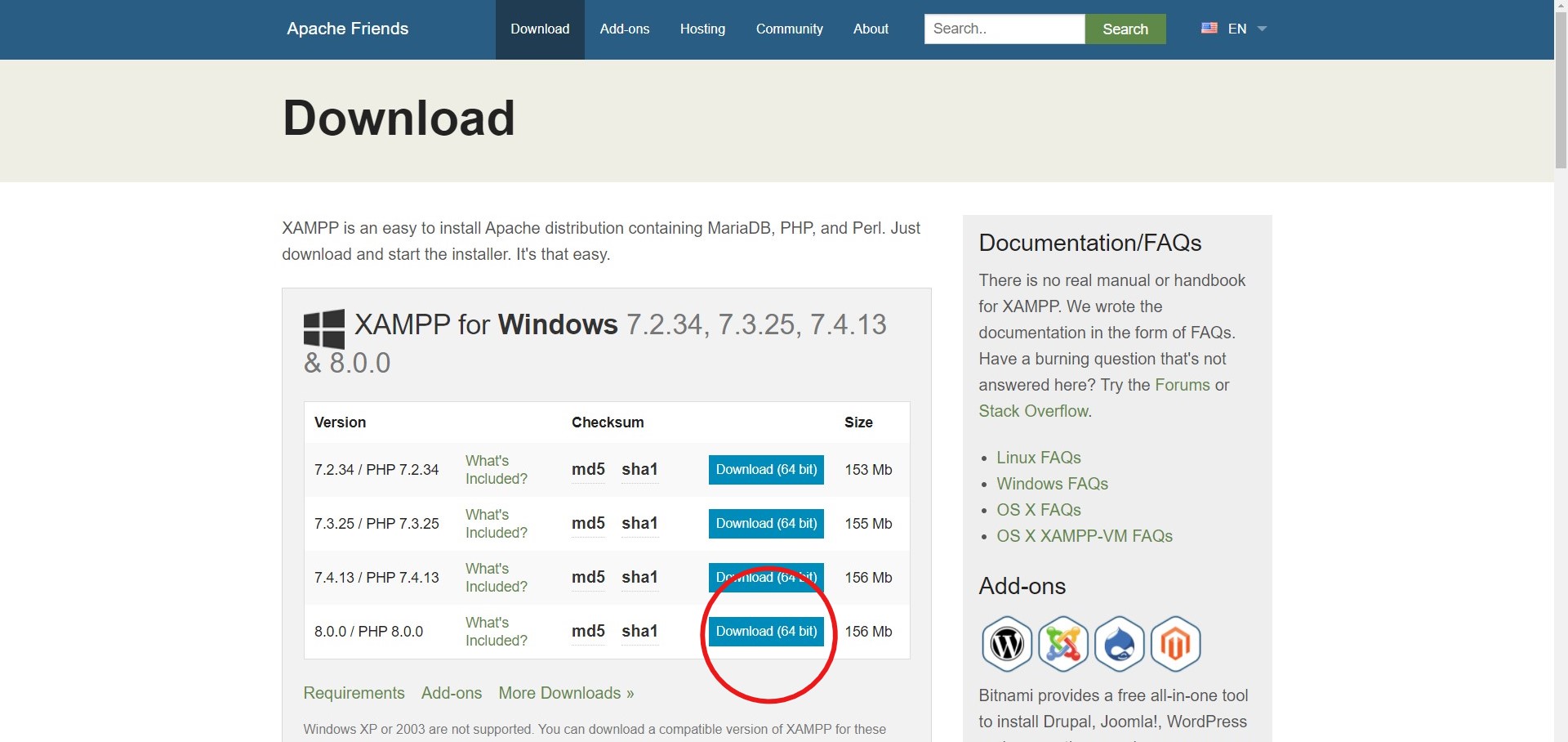
Přehled potřebných programů

1. Balíček XAMPP s PHP.
2. CA certifikát pro psaní do příkazového řádku.
3. Composer pro PHP.
4. Editor PhpStorm s pluginem Symfony.
5. Symfony.
6. WinSCP klient pro Windows.
7. Aplikace GitHub.

Instalace XAMPP s PHP

Nainstalováni XAMPP balíčku je potřebné pro získání jednoduchého vývojového prostředí pro PHP prostředí na vašem počítači.

1. Stáhni XAMPP z <https://www.apachefriends.org/download.html>.



1. Nainstaluj podle instrukcí na svůj počítač.

Obsah obrázku text

Popis byl vytvořen automaticky

Instalace CA certifikátu

CA certifikát je doporučený pro psaní do příkazového řádku. Jde o veřejný digitální certifikát, který je podepsaný veřejným šifrovacím klíčem.

1. Stáhni soubor cacert.pem z <https://curl.haxx.se/docs/caextract.html>.

Obsah obrázku text

Popis byl vytvořen automaticky

1. Vlož do adresáře PHP v XAMPPu (např. C:\xampp\php).

Obsah obrázku stůl

Popis byl vytvořen automaticky

1. V php.ini v sekci [curl] vložit řádek:

curl.cainfo="C:\xampp\php\cacert.pem"

Obsah obrázku text

Popis byl vytvořen automaticky

Instalace Composeru pro PHP

Composer je vhodný pro management závislostí v PHP. Dovoluje deklarovat knihovny, na kterých závisí projekt a dovoluje je také instalovat nebo aktualizovat.

1. Stáhni z <https://getcomposer.org/download/>.

Obsah obrázku text

Popis byl vytvořen automaticky

1. Nainstaluj podle instrukcí na svůj počítač.

Obsah obrázku text

Popis byl vytvořen automaticky

Instalace Symfony

Symfony je PHP framework usnadňující tvorbu webu pomocí vyladěných knihoven

1. Spusť aplikaci ‚Příkazový řádek‘.

Obsah obrázku text

Popis byl vytvořen automaticky

1. Přejdi v příkazovém řádku do adresáře htdocs v XAMPPu (např. C:\xampp\htdocs).



1. Do příkazového řádku poté zadej

php -r "readfile('https://symfony.com/installer');" > symfony

Obsah obrázku text

Popis byl vytvořen automaticky

1. Následně zadej příkaz
   1. php symfony



1. Počkej, než se nainstaluje Symfony Framework.

Instalace WinSCP

WinSCP je SFTP, FTP, WebDAV, S3 a SCP klient sloužící především pro přesunování souboru mezi lokálním počítačem a počítačem vzdáleným na jiném serveru. V našem případě přesun na školní server, na kterém běží naše internetová aplikace.

1. Stáhni z <https://winscp.net/eng/download.php>.

Obsah obrázku text

Popis byl vytvořen automaticky

1. Nainstaluj podle instrukcí na svůj počítač.

Obsah obrázku text

Popis byl vytvořen automaticky

Instalace aplikace GitHub

Aplikace GitHub je velmi populární nástroj pro správu a sdílení kódu aplikace. Je vhodné kód upravovat na Gitu v projektu: <https://github.com/RSP-vspj/Seminarka>.

1. Stáhni z <https://desktop.github.com/> .



1. Spusťte GitHubDesktopSetup.exe a GitHub se nainstaluje.

Spuštění webu na školním serveru

1. Připojte se k vašemu účtu na školním serveru:
   1. 195.113.207.163
2. Přesuňte do složky „/home/user/“ projekt Symfony spravovaný na GitHubu pod názvem „Seminárka“.
3. Vytvořte v „/home/user/public\_html“, který ukazuje do „/home/user/nazev\_webu/web“.
4. Poté ve složce „/home/user/nazev\_webu/web/.htaccess“ zakomntujte „Options -MultiViews“

**Zkrácená verze souboru .htaccess bez komentářů:**

DirectoryIndex app.php

<IfModule mod\_negotiation.c>

</IfModule>

<IfModule mod\_rewrite.c>

RewriteEngine On

RewriteCond %{REQUEST\_URI}::$1 ^(/.+)/(.\*)::\2$

RewriteRule ^(.\*) - [E=BASE:%1]

RewriteCond %{HTTP:Authorization} .

RewriteRule ^ - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]

RewriteCond %{ENV:REDIRECT\_STATUS} ^$

RewriteRule ^app\.php(?:/(.\*)|$) %{ENV:BASE}/$1 [R=301,L]

RewriteCond %{REQUEST\_FILENAME} -f

RewriteRule ^ - [L]

RewriteRule ^ %{ENV:BASE}/app.php [L]

</IfModule>

<IfModule !mod\_rewrite.c>

<IfModule mod\_alias.c>

RedirectMatch 302 ^/$ /app.php/

</IfModule>

</IfModule>

1. Pak upravte práva tak, aby Apache mohl zapisovat do cache, logu a sessions – ve složce „/home/user/nazev\_webu/var“: „chmod -R o+w logs“, „chmod -R o+w cache“, „chmod -R o+w sessions“, popřípadě je možné to upravit pro celou složku var.
2. Nyní zakomentujte „/home/user/nazev\_webu/web/app\_dev.php“ ověřování IP adresy, aby se bylo možné dostat i na developerské prostředí.

**Ukázka souboru app\_dev.php**

<?php

use Symfony\Component\Debug\Debug;

use Symfony\Component\HttpFoundation\Request;

// If you don't want to setup permissions the proper way, just uncomment the following PHP line

// read https://symfony.com/doc/current/setup.html#checking-symfony-application-configuration-and-setup

// for more information

umask(0071);

// This check prevents access to debug front controllers that are deployed by accident to production servers.

// Feel free to remove this, extend it, or make something more sophisticated.

/\*if (isset($\_SERVER['HTTP\_CLIENT\_IP'])

|| isset($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])

|| !(in\_array(@$\_SERVER['REMOTE\_ADDR'], ['127.0.0.1', '::1'], true) || PHP\_SAPI === 'cli-server')

) {

header('HTTP/1.0 403 Forbidden');

exit('You are not allowed to access this file. Check '.basename(\_\_FILE\_\_).' for more information.');

}\*/

require \_\_DIR\_\_.'/../vendor/autoload.php';

Debug::enable();

$kernel = new AppKernel('dev', true);

if (PHP\_VERSION\_ID < 70000) {

$kernel->loadClassCache();

}

$request = Request::createFromGlobals();

$response = $kernel->handle($request);

$response->send();

$kernel->terminate($request, $response);

1. Nyní již naleznete web na adresách:

<https://alpha.kei.vspj.cz/~user/nazev_webu/>

[https://alpha.kei.vspj.cz/~user/nazev\_webu/app dev.php](https://alpha.kei.vspj.cz/~user/nazev_webu/app%20dev.php)

Konfigurace uživatelských rolí

Uživatelské role jsou realizovány pomocí frameworku Symfony a jsou nastavny v souboru \app\config\security.yml.

Seznam uživatelů si načítá z tabulky *uzivatel* a k přihlašování slouží atributy *login* a *heslo*, uživatelské role jsou definovány atributem *role.* Hesla jsou šifrována pomocí algoritmu.

**Atribut *role* může nabývat těchto hodnot:**

0: ROLE\_ADMIN – administrátor

1: ROLE\_UZIVATEL – uživatel

2: ROLE\_REDAKTOR – redaktor

3: ROLE\_RECENZENT – recenzent

4: ROLE\_SEFREDAKTOR – šéfredaktor

5: ROLE\_NEREG\_AUTOR – neregistrovaný autor

Uživatele mohou registrovat a upravovat pouze uživatelé s oprávněním administrátor, redaktor a šéfredaktor. Formuláře na správu tabulky *uzivatel* jsou po přihlášení dostupné na routě /uzivatel.

Pokud je tabulka *uzivatel* prázdná, je nejprve nutné přímo v MySQL vytvořit uživatele s oprávněním administrátor (atribut *role* s hodnotou 0) a do atributu *heslo* vložit hash zakódovaného hesla pomocí algoritmu bcrypt. Hash je možné získat např. pomocí <https://bcrypt-generator.com>.

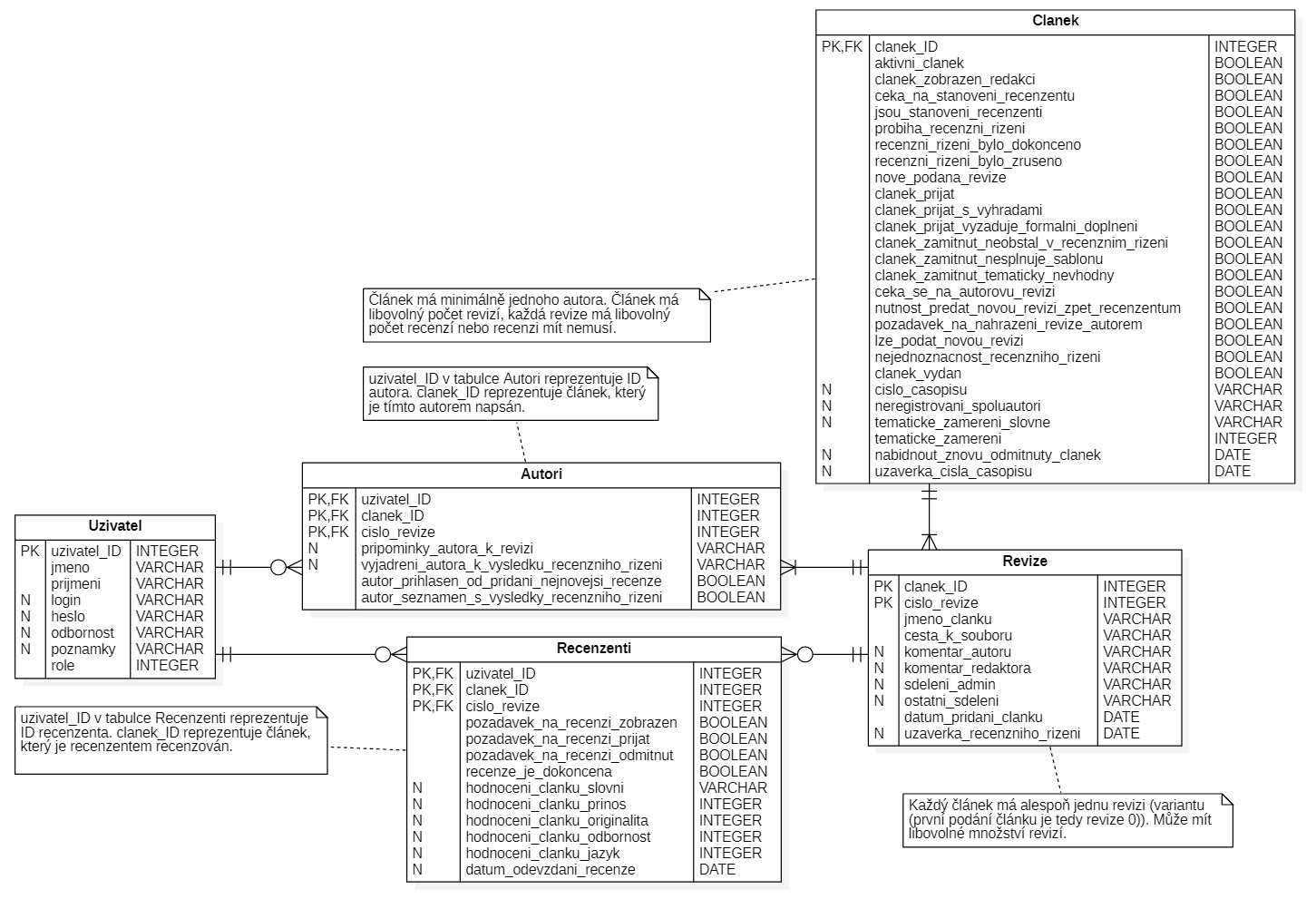
Pokud je nový uživatel registrován pomocí formuláře *new*, zadává se heslo v nešifrované podobě, do databáze se následně uloží jeho hash.

Pokud je uživatel upravován pomocí formáláře *new*, je nutné zadat vždy znovu heslo v nešifrované podobě, v opačném případě dojde při uložení hodnot k zašifrování stávajícího hashe, který tak již nebude původnímu heslu odpovídat.

Konfigurace implementace databáze

Databáze je implementována v rámci frameworku Symfony pomocí sady PHP knihoven Doctrine.

**Implementace vychází z toho relačního modelu:**



Z každé tabulky je za pomoci příkazu *# php app/console doctrine:generate:entity* vygenerován objektový model reprezentovaný těmito soubory v adresáři src/AppBundle/Entity:

* Autori.php
* Clanek.php
* Recenzenti.php
* Revize.php
* Uzivatel.php

Následně jsou za pomoci příkazu *# php app/console doctrine:schema:update --force* na základě těchto objektových modelů vygenerovány příslušné tabulky v databázi MySQL.

**Současný stav implementace:**

1. Všechny tabulky jsou vygenerovány a vytvořeny v databázi.
2. Tabulka Uzivatel současně slouží pro registraci a autentizaci uživatelů pomocí konfiguračního souboru *security.yml*.
3. Oproti relačnímu modelu je prvním sloupcem každé tabulky automaticky inkrementovaný sloupec *id*, které je vždy primárním klíčem. Původně navržené řešení působilo problémy při snaze vytvářet relace. Došlo tak těmto změnám:
4. tab. ***Autori***: přidán primární klíč **id**
5. tab. ***Clanek:* clanek\_ID** změněno na **id** (primární klíč)
6. tab. ***Recenzenti:*** přidán primární klíč **id**
7. tab. ***Revize***: **cislo\_revize** změněno na **id** (primární klíč)
8. tab. ***Uzivatel***: **uzivatel\_ID** změněno **id** (primární klíč)
9. Zatím se nepodařilo v objektovém modelu Symfony vytvořit fungující relace se složenými klíči a cizími primárními klíči podle výše uvedeného relačního modelu databáze. Proto v tuto chvíli nejsou složené klíče a cizí primární klíče implementovány.
10. Jsou implementovány tabulky ***Uzivatel*** a ***Clanek***, aby mohla být zprovozněna alespoň základní funkcionalita zadávání a evidence článků. Tabulky jsou spojeny relací 1:N. Do tabulky ***Clanek*** byly přidány tyto atributy:
    1. jmeno\_clanku (VARCHAR)
    2. cesta\_k\_souboru (VARCHAR)
    3. uzivatel\_ID (FK, Uzivatel, id)

Pro každou tabulku byl pro usnadnění práce s daty vygenerován CRUD pomocí nástroje petkopara. Formuláře pro vkládání, úpravu a mazání dat se nacházejí pod routami:

/autori

/clanek

/recenzenti

/revize

/uzivatel

Správa na webové stránce

Toto je základní uživatelské rozhraní dostupné pro uživatele. Skrze přihlašovací formulář v horní liště vpravo se přihlašuje i samotný administrátor. Získá tím přístup k možnostem, které má administrátorský účet.

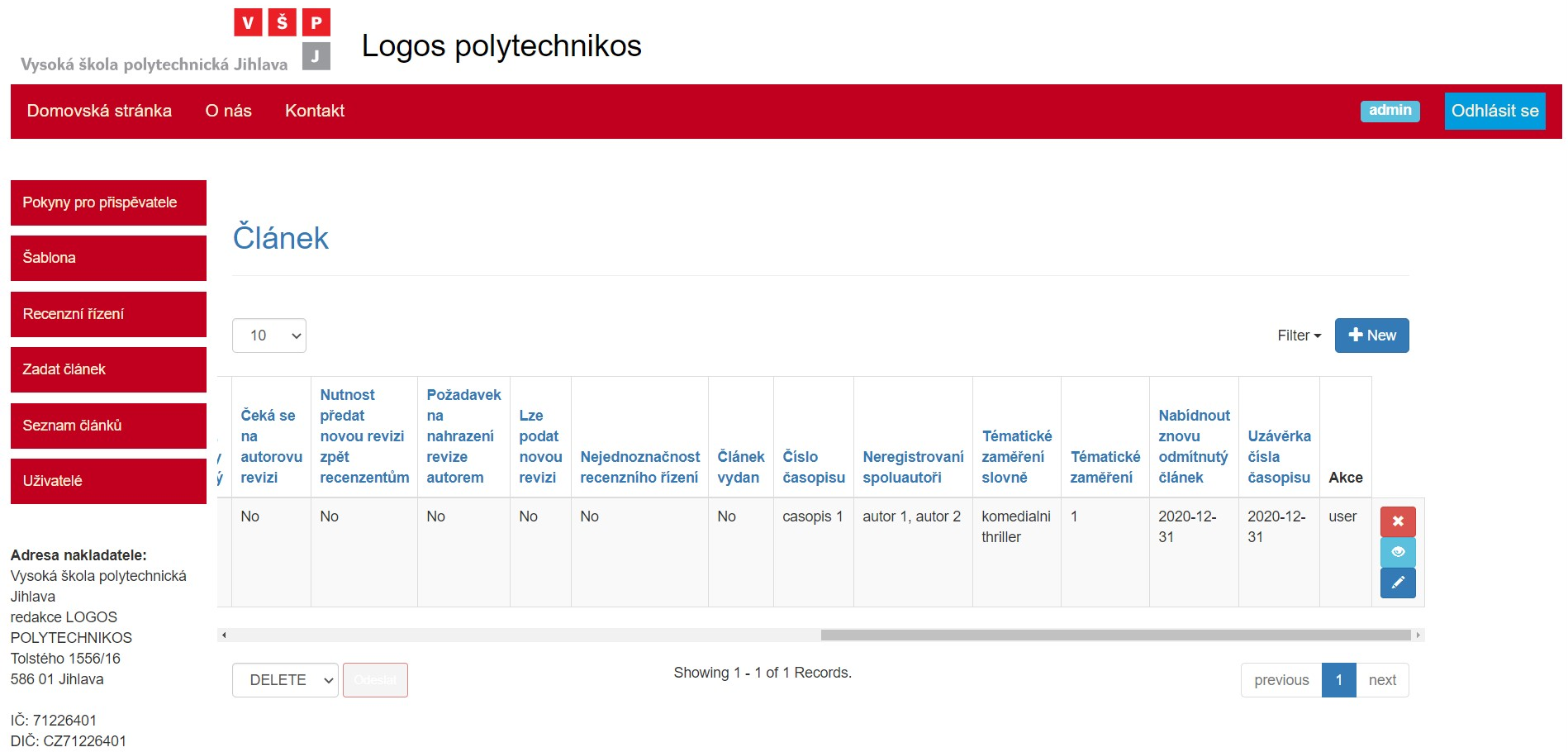


Běžný defaultní přístup je dán konfigurací uživatelských rolí (str. 12).

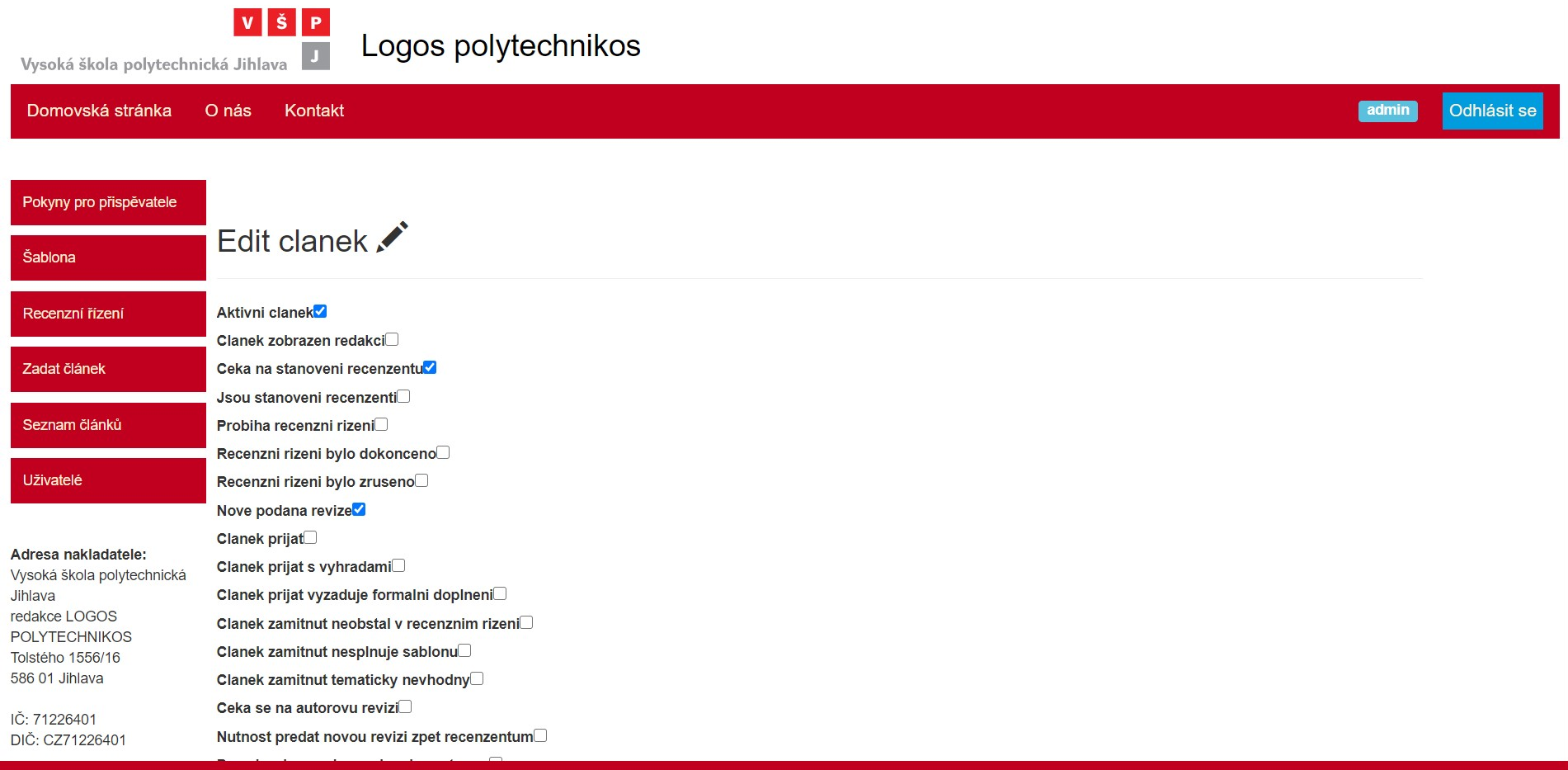


Po přihlášení na administrátorský účet se zpřístupní dvě další možnost v levém Menu. Jde o „Seznam článků“ a „Uživatelé“. V menu je po kliknutí na „Seznam článků“ možné články editovat a odstraňovat, případně pouze ukázat jejich aktuální stav co se týče např. stanovení recenzentů, přijetí článku apod. Je také možné přidat nový článek. Skrze kliknutí v menu na „Uživatelé“ je možné taktéž uživatele editovat, odstraňovat, kontrolovat a přidávat uživatele nové.

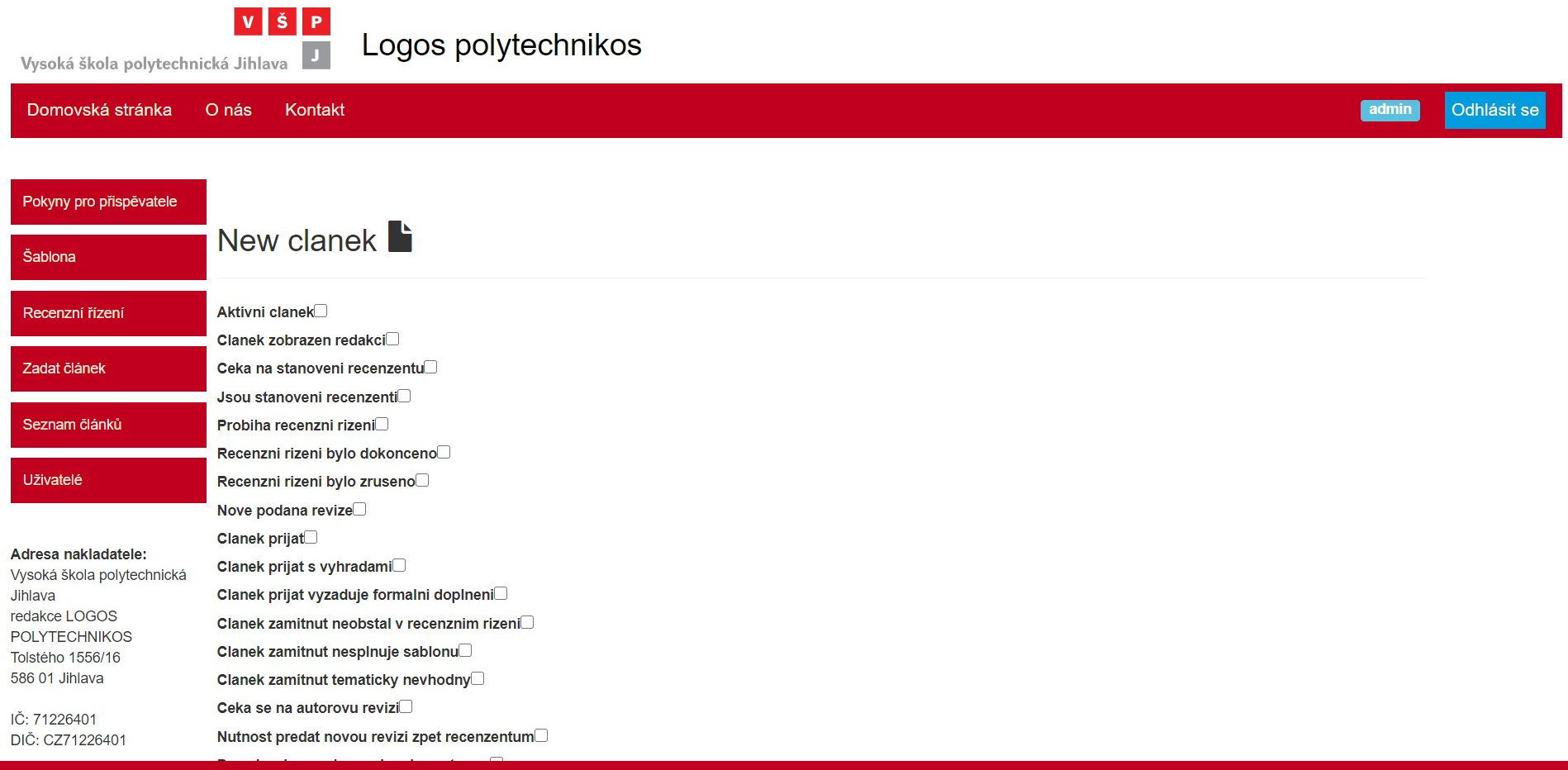
**Zde je vidět správa všech článků z pohledu administratora:**



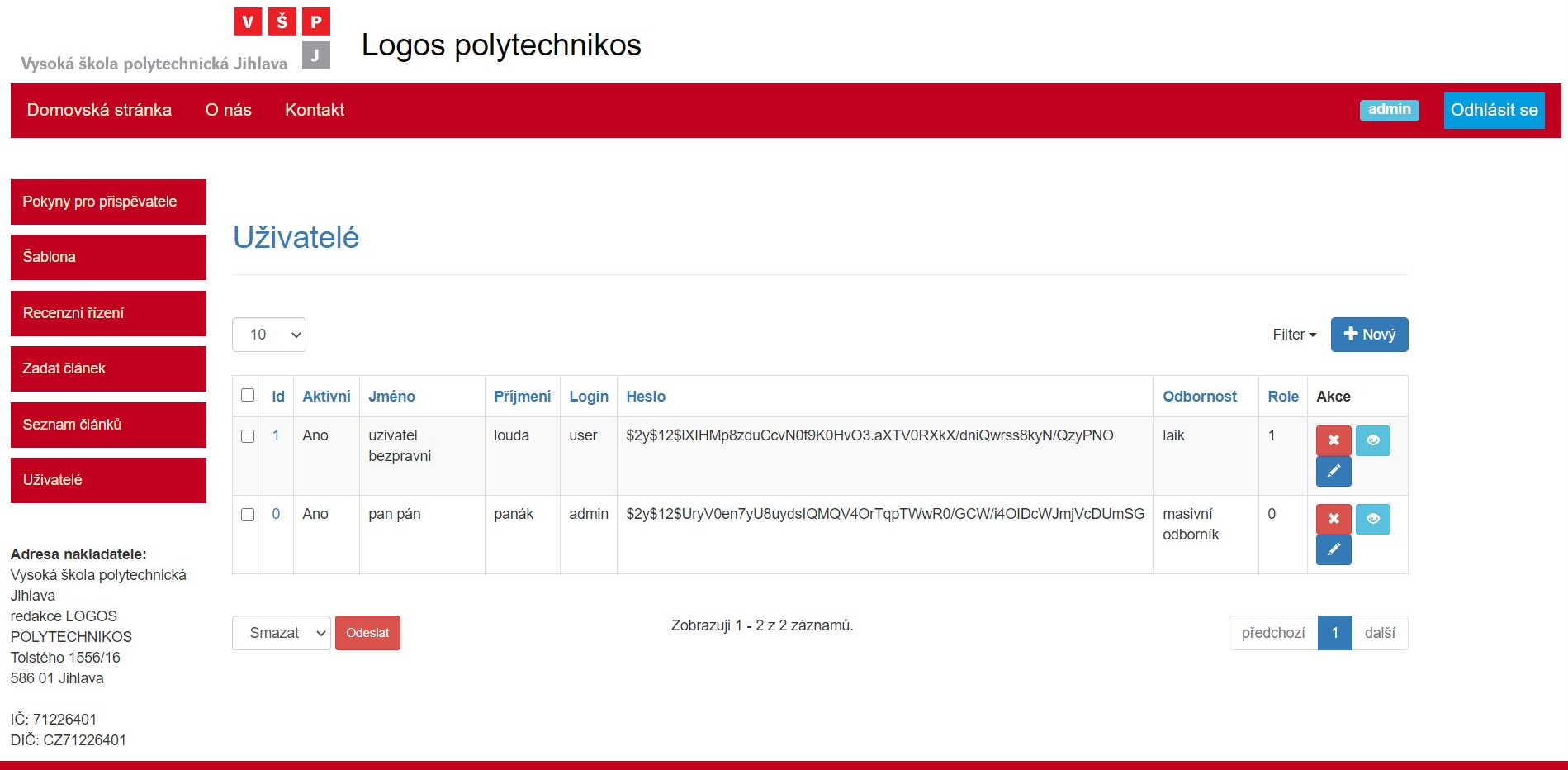
**Články je možné editovat pomocí kliknutí na tlačítko „Edit“ u článku:**

****

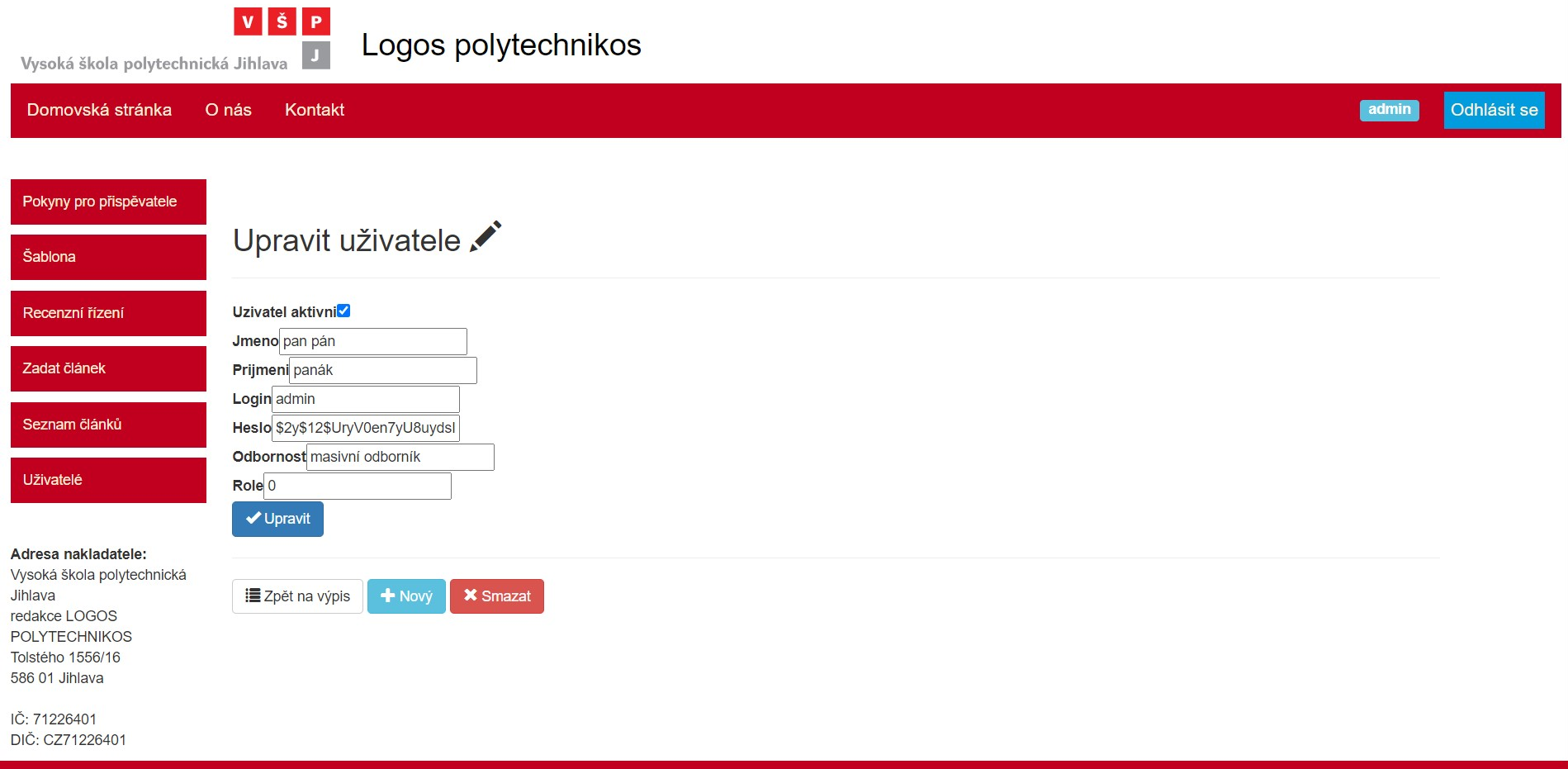
**Pomocí tlačítka „New“ je možnost přidávat nové články:**

****

**Na stránce „Uživatelé“ je vidět administrátorská správa všech uživatelů:**

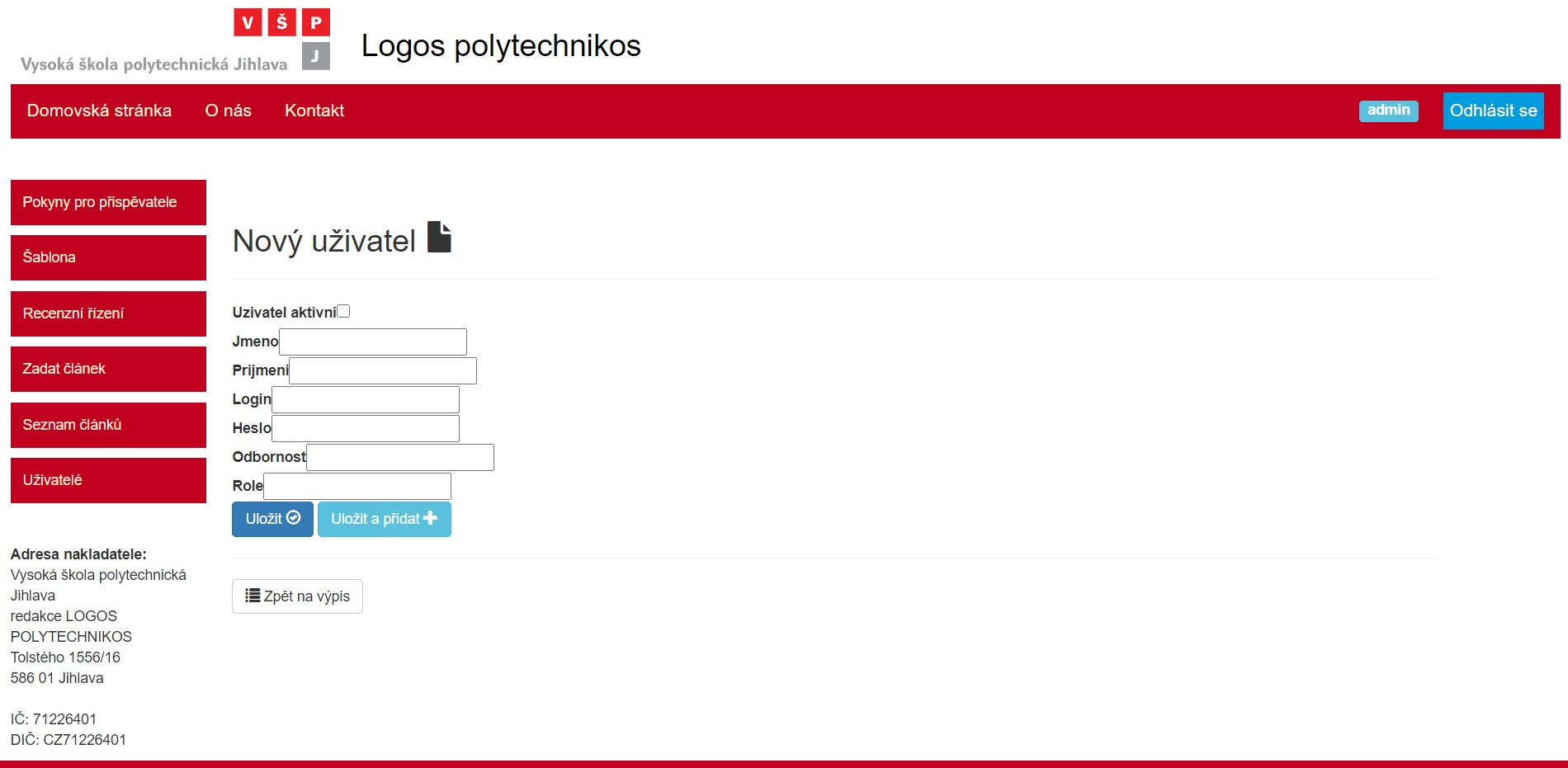
****

**Je možné uživatele upravovat pomocí tlačítka „Edit“:**

****

Zde je důležité připomenout, že heslo je zde zadáváno pomocí hashe vysvětleným podrobněji na straně 12.

**Lze také uživatele přidávat skrze tlačítko „Nový“:**



Podpora

Poskytovatel

Team Tarzani

Pavel Kincl Product Owner

Václav Seidler Scrum Master

Štefan Pilát Team Member

Jan Přibyl Team Member

Jakub Štulíř Team Member

Jan Karas Team Member

**Kontakty:**

|  |
| --- |
|  |
| **Telefon** 567 141 111  **E-mail** vspj@vspj.cz |

© Team Tarzani 2021