

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE**

Evidenčné číslo: FCHPT-5415-50946

**WEBOVÝ PORTÁL SLOVENSKEJ SPOLOČNOSTI
PRIEMYSELNEJ CHÉMIE**

BAKALÁRSKA PRÁCA

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE**

**WEBOVÝ PORTÁL SLOVENSKEJ SPOLOČNOSTI
PRIEMYSELNEJ CHÉMIE**

BAKALÁRSKA PRÁCA

FCHPT-5415-50946

Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo

Vedúci záverečnej práce/školiťel': Ing. Ľuboš Čirka, PhD.

Bratislava, 2014

Igor Stupavský

Zadanie

Slovenská technická univerzita v Bratislave
Ústav informatizácie, automatizácie a matematiky

Fakulta chemickej a potravinárskej technológie
Akademický rok: 2013/2014
Evidenčné číslo: FCHPT-5415-50946



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Igor Stupavský**
ID študenta: 50946
Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve
Kombinácia študijných odborov: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo
Vedúci práce: Ing. Ľuboš Čírka, PhD.

Názov práce: **Webový portál Slovenskej spoločnosti priemyselnej chémie**

Špecifikácia zadania:

Bakalárska práca sa venuje tvorbe webového portálu Slovenskej spoločnosti priemyselnej chémie. Portál je postavený na redakčnom systéme.

Úlohy:

1. Rešerš dostupných redakčných systémov.
2. Naštudovať jazyky PHP a SQL, zoznámiť sa s databázou MySQL.
3. Naštudovať architektúru zvoleného redakčného systému.
4. Nainštalovať a nakonfigurovať redakčný systém.
5. Vykonať analýzu a navrhnúť štruktúru nového modulu.
6. Programovo realizovať modul.
7. Implementovať vytvorený modul do redakčného systému.

Rozsah práce: 30

Zoznam odbornej literatúry:

1. Buytaert, D. Drupal. [online]. 2011. URL: <http://drupal.org/>.
2. Polzer, J. *Drupal : Podrobný průvodce tvorbou a správou webů*. Brno: Computer Press, 2008. 262 s.
3. Polzer, J. *333 tipů a triků pro Drupal*. Brno: Computer Press, 2010. 264 s. ISBN 978-80-251-2942-5.

Riešenie zadania práce od: 17. 02. 2014

Dátum odovzdania práce: 24. 05. 2014

L. S.

Igor Stupavský
študent

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

Ďakujem vedúcemu práce
Ing. Ľubošovi Čirkovi, PhD. za jeho
čas, pripomienky a pomoc pri vypracovaní
tohto bakalárskeho projektu.

Abstrakt

V bakalárskej práci sa venujem tvorbe webového prostredia pre Slovenskú spoločnosť priemyselnej chémie (SSPCH) pre potreby jej členskej základne a odbornej i laickej verejnosti. V teoretickej časti práce sa venujem predkladám ucelený prehľad informácií o jednotlivých použitých technológiách. Následne sa venujem základnej administratíve a modulárnemu systému Drupal. Praktická časť práce sa zaoberá tvorbou webovej stránky SSPCH a tvorbou modulu vhodného pre potreby štandardného spravovania stránky vedením danej spoločnosti.

Kľúčové slová: Drupal; modul; programovanie

Abstract

This bachelor thesis deals with development of an internet interface for Slovak Society for Industrial Chemistry (SSPCH), in accordance with needs of its management, members, professionals as well as non-professionals. The theoretical part is an integral overview of the technologies used. Next, it deals with the basic administration and modular system of Drupal. Practical part deals with the creation of a website for SSPCH and with creation of a module suitable for the needs of a standard website administration, conducted by the management of the society.

Key words: Drupal; Modul; Programming

Obsah

Zoznam ilustrácií	12
Zoznam tabuliek	13
Úvod	14
1 CMS systémy.....	15
1.1 História drupalu	16
1.2 Pred inštaláciou.....	17
1.3 Štruktúra Drupal-u.....	18
1.3.1 Témy a preklady	18
1.3.2 Moduly.....	19
2 Tvorba stránky SSPCH.....	21
3 Tvorba vlastného modulu	22
3.1 Súbory modulu	22
3.1.1 Súbor tabulka.info	23
3.1.2 Súbor tabulka.install	24
3.1.3 Súbor tabulka.module	27
3.2 Budúcnosť modulu	32
4 Budúcnosť stránky SSPCH	33
5 Záver.....	34
Zoznam použitej literatúry.....	35
Prílohy	36

Zoznam ilustrácií

Obr. 1	Základná obrazovka MAMP-u	17
Obr. 2	Umiestnenie modulu a jeho štruktúra.....	23
Obr. 3	Vytvorená tabuľka.....	27
Obr. 4	Formulár	30
Obr. 5	Výsledná tabuľka.....	31

Zoznam tabuliek

Tab. 1	Najznámejšie CMS systémy.....	15
--------	-------------------------------	----

Úvod

Každý človek na tomto svete, či už jednoduchý občan, verejne činná osoba cíti potrebu dať do sveta o sebe vedieť, o svojich myšlienkach, nápadoch, skúsenostiach. Väčšina sa prezentuje pomocou sociálnych sietí, niektorí cez svoje vlastné internetové stránky. Každá firma, spoločnosť ide za svojimi zákazníkmi, členmi. Preto nie je žiaden div, že väčšina (ak nie všetky) majú svoje vlastné miesto v elektronickom svete nazývanom internet. Firmy ktoré obchodujú s reálnym tovarom chcú aby bolo o ich tovare, aktivitách vedieť a preto zakladajú rôzne e-shopy a firemné stránky pre svojich zákazníkov. Spoločnosti, ktoré nič nepredávajú, vyvíjajú aktivity, rôzne stretnutia na odborné témy, prednášky, diskusie majú potrebu dať o sebe, mieste konania a čase vedieť svojim členom, prípadne ostatným záujemcom.

Práve pre tieto potreby sme boli oslovení Slovenskou spoločnosťou priemyselnej chémie (SSPCH) na tvorbu ich internetovej stránky. Požiadavka bola jasne zadefinovaná, avšak výber vhodných technológií ako aj grafického či presnejšieho obsahového naplnenia, podliehalo našej úvahe a schopnostiam. Pustili sme sa teda do hľadania najvhodnejšej metódy práce pre náš projekt.

Spočiatku sme uvažovali nad klasickým spôsobom programovania internetu pomocou zdrojových kódov. Tento spôsob je však pracný. Začali sme sa teda ohliadať po príjemnejšom spôsobe a môj školiteľ upriamil našu pozornosť na jeden z CMS systémov, ktorý sa volá Drupal.

1 CMS systémy

CMS systém (Content Management System – systém pre správu obsahu) je webový redakčný systém, ktorý slúži na vytvorenie a editáciu webových stránok i bez podpory aktívnej programátorskej gramotnosti. Prioritne, ako už názov napovedá boli myslené ako redakčné systémy, kde existuje potreba pravidelného pridávania a aktualizovania obsahu (článkov...). V redakciách, kde je veľký počet užívateľov (redaktorov, fotografov, dizajnérov...) a nie je nevyhnutne potrebné pre ich prácu učiť sa programovací jazyk.

Systémy ktoré patria do tejto kategórie rozlišujeme do dvoch základných druhov: platené, alebo neplatené (free). Využívajú sa na tvorbu obsahu prostredie webového prehliadača a pracujú s rôznymi programovacími jazykmi (PHP, XHTML, JavaScript...), v súčinnosti s databázami.

Zoznam jednotlivých CMS systémov sa nachádza na stránke <http://cmsmatrix.org/> (cca 1200, údaj k 24. 5. 2013). Avšak, ani tento výpočet ešte nemusí byť konečný a neustále na tomto segmente trhu pôsobí neviditeľná ruka trhu. Niektoré vznikajú a nepotrebné, zastarané či prekonané systémy postupne zanikajú (rovnako aj podpora pre ne).

Ja osobne som si vybral z veľkého množstva týchto CMS systém Drupal. Najskôr mi ho odporúčal školiť. Neskôr keď som sa s ním postupne zoznamoval, zistil som sám jeho výhody (hlavne rýchlu prácu a možnosť vložiť dodatočne moduly od tretích strán ktoré systém rozširujú a spôsobujú jeho značnú flexibilitu). Každému však môže vyhovovať iný systém. Ak by boli požiadavky a zameranie SSPCH iné, napríklad potreba založiť e-shop na predaj svojich výrobkov, asi by siahol napríklad po CMS systéme Magento či iných.

V nasledujúcej tabuľke uvádzam najznámejšie a najpoužívanéjšie CMS systémy.

Tab. 1 Najznámejšie CMS systémy

Názov CMS	Popis	Zdroj na stiahnutie
Joomla	Z pohľadu užívateľa univerzálny.	http://www.joomla.org/download.html
Drupal	Najideálnejšia voľba pre všetkých vývojárov.	http://drupal.org/download
WordPress	Najdynamickejšie sa rozvíjajúci.	http://sk.wordpress.org/

Magento	Tvorba elektronickej predajne.	http://www.magentocommerce.com/download
TYPO3	Rozšírení v Európe a stále ľahšia administrácia.	http://typo3.org/download/

1.1 História drupalu

Drupal ako redakčný systém vznikol v roku 2000 naprogramovaným študenta D. Buytaert-a, kôli potrebe zosieťovania dvoch vedľajších izieb na internáte. O pripojenie na internet sa delilo dokopy 8 študentov. Vznikla potreba u týchto „užívateľov“, aby si zanechávali medzi sebou krátke odkazy a tak isto aj miesto pre diskusiu. Tak sa aj stalo a Dries naprogramoval jednoduchý web so zabudovaným diskusným fórom ktorý začali študenti používať.

Po škole vznikla otázka: „Ako sa budeme navzájom informovať o novinkách a ako ostať v kontakte? “ Svojmu systému dali názov Drop a zaregistrovali si doménu drop.org ktorá bola vtedy voľná. Slovo Drop je belgického pôvodu a znamená dedina.

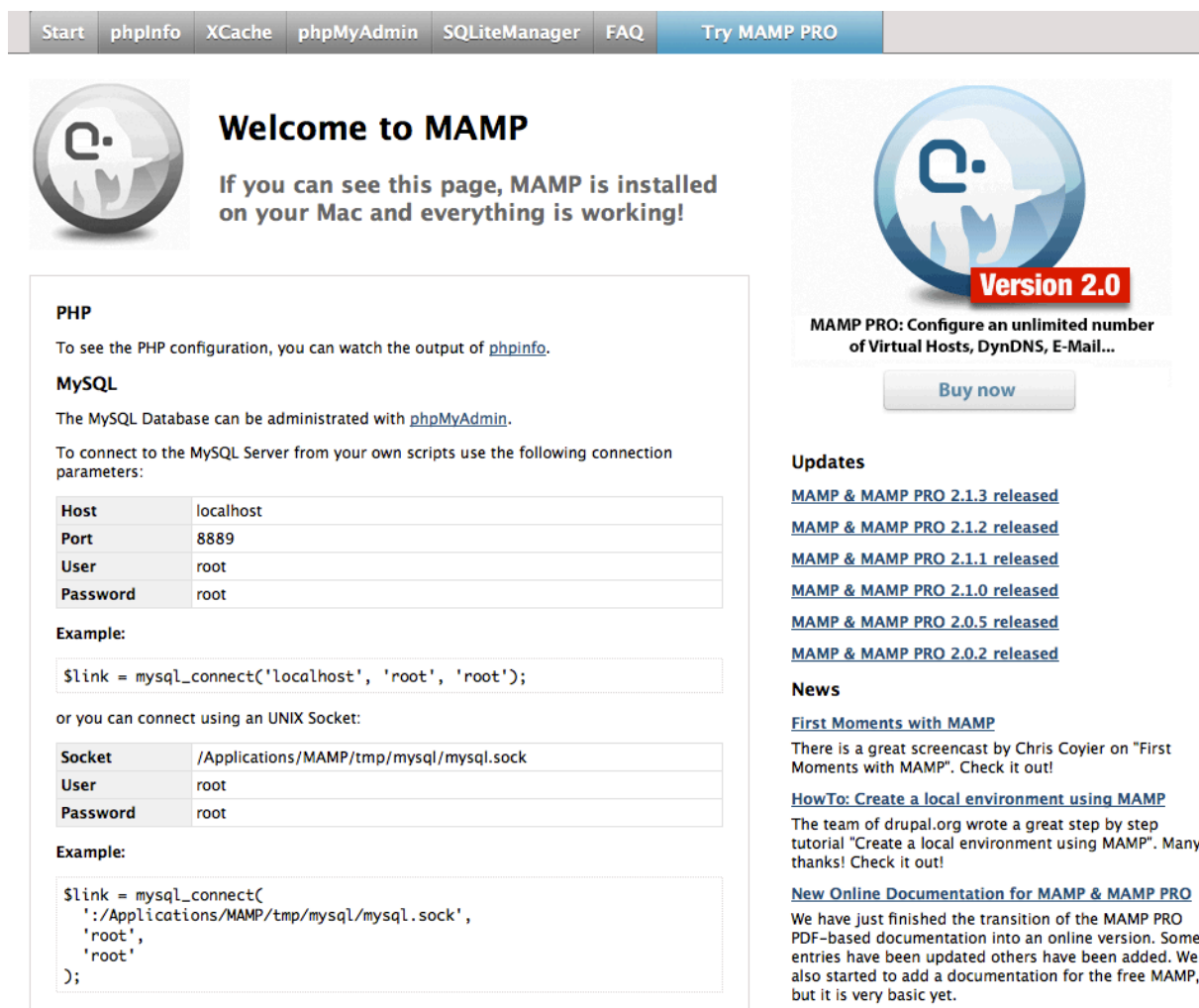
Základný skelet systému bol hotový a postupne sa doň pridávali jednotlivé vylepšenia. Ako napríklad moderovanie, hodnotenie obsahu a podobne. Tieto diskusie sa stali hnacím motorom vývoja celého systému známeho pod menom Drupal.

V roku 2001 bol systém uvoľnený pre širokú verejnosť k testovaniu a pridávaniu nových funkcií. Meno Drupal vzniklo z pôvodného Drop, čo bola anglická výslovnosť „druppel“ – kvapka.

Po krátkom čase sa začalo Drupalu venovať stále viac a viac ľudí, dedinka sa utešene rozrastala. Postupne sa z nej stávalo mestečko, mesto a nakoniec sa stala jednou z najväčších komunít vo svete redakčných systémov. Kde je v súčasnej dobe prihlásených približne 330 000 užívateľov z celého sveta. Tým o 130 000 užívateľov predčil druhý CMS systém v rade Joomla. [1]

1.2 Pred inštaláciou

Pri hľadaní vhodného servera, ktorý by som si nainštaloval do počítača, som bral do úvahy typ operačného systému, s čím pracuje (phpMyAdmin), veľkosť základného balíčku a jeho cenu (bezplatnosť). Voľba padla na server s názvom: MAMP (aktuálna verzia dostupná na <http://www.mamp.info/en/downloads/> (údaj k 20. 5. 2014)). V základnej zip verzii s veľkosťou cca 140 MB. Po nainštalovaní do počítača budete vidieť základnú obrazovku serveru MAMP, ktorej ukážku som priložil ako obr. 1.



Start phpInfo XCache phpMyAdmin SQLiteManager FAQ **Try MAMP PRO**

Welcome to MAMP

If you can see this page, MAMP is installed on your Mac and everything is working!

PHP

To see the PHP configuration, you can watch the output of [phpinfo](#).

MySQL

The MySQL Database can be administrated with [phpMyAdmin](#).

To connect to the MySQL Server from your own scripts use the following connection parameters:

Host	localhost
Port	8889
User	root
Password	root

Example:

```
$link = mysql_connect('localhost', 'root', 'root');
```

or you can connect using an UNIX Socket:

Socket	/Applications/MAMP/tmp/mysql/mysql.sock
User	root
Password	root

Example:

```
$link = mysql_connect(
    ':/Applications/MAMP/tmp/mysql/mysql.sock',
    'root',
    'root'
);
```

Version 2.0

MAMP PRO: Configure an unlimited number of Virtual Hosts, DynDNS, E-Mail...

[Buy now](#)

Updates

- [MAMP & MAMP PRO 2.1.3 released](#)
- [MAMP & MAMP PRO 2.1.2 released](#)
- [MAMP & MAMP PRO 2.1.1 released](#)
- [MAMP & MAMP PRO 2.1.0 released](#)
- [MAMP & MAMP PRO 2.0.5 released](#)
- [MAMP & MAMP PRO 2.0.2 released](#)

News

[First Moments with MAMP](#)

There is a great screencast by Chris Coyier on "First Moments with MAMP". Check it out!

[HowTo: Create a local environment using MAMP](#)

The team of drupal.org wrote a great step by step tutorial "Create a local environment using MAMP". Many thanks! Check it out!

[New Online Documentation for MAMP & MAMP PRO](#)

We have just finished the transition of the MAMP PRO PDF-based documentation into an online version. Some entries have been updated others have been added. We also started to add a documentation for the free MAMP, but it is very basic yet.

Obr. 1 Základná obrazovka MAMP-u

Následne je nutné v ponuke prejsť na phpMyAdmin a vytvoriť si databázu, do ktorej sa budú zapisovať jednotlivé tabuľky systému Drupal.

Nesmieme si zabudnúť naštudovať základy písania kódu pre systém Drupal (jeho zvláštnosti), jazyka PHP a SQL.

1.3 Štruktúra Drupal-u

Drupal je napísaný v jazyku PHP a šírený pod licenciou GNU GPL. Skladá sa z jadra, tém a používateľských modulov dostupných na webovej stránke drupal.org. Moduly je možno stiahnuť aj od iných používateľov, ale tieto sú podporované a ich aktuálnosť sa pravidelne kontroluje. [2]

Je tu možnosť si napísať v prípade potreby aj vlastný modul, keďže je štruktúra modulu dobre známa. Ale tvorba modulu si vyžaduje už určité programátorské zručnosti, a preto si modul bežný užívateľ vytvárať pravdepodobne nebude.

Rovnako je možné si vytvoriť aj vlastnú tému vzhľadu, k čomu však tiež budeme potrebovať znalosť jazykov PHP a CSS.

Súčasťou mojej práce je aj tvorba jednoduchého modulu, ktorý som implementoval do tvorenej internetovej stránky

1.3.1 Témy a preklady

Nami vytvorené/stiahnuté témy sa ukladajú do *sites/all/themes*, nie priamo do *themes*, lebo tu sú umiestnené základné témy, ktoré patria do jadra a sú v prípade upgrade systému prepisované.

Do priečinku *profiles/standard/translations* môžeme umiestniť preklady do rôznych jazykov, v našom prípade do slovenčiny. V jazyku okrem angličtiny, ktorého preklad tu máme uložený bude s nami vedieť Drupal komunikovať v administračnom paneli i inde v systéme. Preklady majú koncovku *.po*.

V *scripts* sa nachádzajú *.sh* skripty k chodu celého systému a v *misc* sú uložené jednotlivé obrázky a ikony aj s JavaScriptami.

1.3.2 Moduly

Drupal je systém zložený z modulov a to mu dáva veľkú flexibilitu a možnosť pre každého užívateľa si doladiť takmer čokoľvek podľa svojich predstáv.

Základné rozdelenie modulov je na:

- a) Moduly ktoré využíva jadro (v zložke *modules*)
- b) Rozširujúce moduly ktoré nie sú súčasťou jadra (v zložke *sites/all/modules*)

a) Moduly ktoré využíva jadro

- Tieto moduly sú základné a všeobecne platí dohovor, že sem svoje stiahnuté, či vytvorené moduly neumiestňujeme. Ak by sme ich sem z akéhokoľvek dôvodu aj umiestnili, fungovali by rovnako ale v súčasnosti na naše moduly máme osobitnú zložku.
- Základné moduly, ktoré som mal vo svojej inštalácii Drupal-u (ich zoznam sa dá zistiť v danom priečinku, alebo v programe v Moduly - Jadro).

b) Rozširujúce moduly, ktoré nie sú súčasťou jadra

- Tieto moduly sú rozširujúce a aj kôli nim sa stal Drupal tak obľúbeným a rozšírením systémom.
- Umožňujú jeho ohromnú flexibilitu a prispôsobiteľnosť skoro každému používateľovi.
- Rozdeľujeme ich na:
 - ba) Moduly z oficiálnej stránky Drupalu
 - bb) Vlastné naprogramované moduly
 - bc) Neoficiálne moduly tretích strán

ba) **Moduly z oficiálnej stránky Drupalu**

- Užívateľ sám pri tvorbe svojich stránok príde na to, že niekedy potrebuje použiť funkciu ktorú mu samotné jadro Drupalu neposkytuje a preto začína hľadať ako by daný problém mohol vyriešiť.

- Na tieto prípady slúžia oficiálne stránky Drupalu kde si môžete stiahnuť skoro 27 000 modulov (údaj k 20. 5 2014), ktoré vytvorili samotný používatelia a sú overené a pravidelne aktualizované. [3]

bb) **Vlastné naprogramované moduly**

- Ak by sme nenašli vhodný modul ani po dlhšom hľadaní alebo by sme si z rôznych iných dôvodov chceli vytvoriť modul sami, aj túto možnosť nám poskytuje tento redakčný systém.
- Na prvý pohľad by sa nám mohla zdať tvorba vlastného modulu ako ťažká vec, ale ak človek pochopí základne princípy ich tvorby a vie aspoň základy programovania v PHP, HTML a SQL, nie je to až tak zložité.
- Dôležité je si svoj problém pomenovať a analyzovať čo od svojho modulu vlastne očakávame že bude robiť.
- Postup a potrebné informácie vysvetlím v kapitole 3. (Tvorba vlastného modulu).

bc) **Neoficiálne moduly tretích strán**

- Tieto moduly nie sú oficiálne podporované a sú na vlastnú zodpovednosť.

2 Tvorba stránky SSPCH

Pri tvorbe internetovej stránky Slovenskej spoločnosti priemyselnej chémie sme si museli najskôr dôkladne analyzovať požiadavky.

Analýza požiadaviek:

- 1) Bolo treba vytvoriť webovú stránku spoločnosti, ktorá by bola estetická.
- 2) Zároveň musela byť používateľsky príjemná.
- 3) Čo najviac prehľadná a flexibilná.
- 4) Musela umožniť prezeranie informácií neprihláseným členom.
- 5) Bolo potrebné vytvoriť ucelenú databázu členskej základne.

Z týchto požiadaviek nám vyplynulo, že je treba spraviť nasledujúce kategórie užívateľov, pričom pre každú kategóriu boli nastavené rôzne oprávnenia:

- a) Neprihlásený používateľ
- b) Prihlásený používateľ – člen
- c) Prihlásený používateľ – vedenie
- d) Prihlásený používateľ – admin

V prípade potreby je možnosť blokovania jednotlivých IP adries, ak by boli podozrivé, alebo by sa z nich niekto pokúšal dostať neoprávnenie do systému.

3 Tvorba vlastného modulu

Po úvahe, aký modul by bolo treba pre potreby SSPCH vytvoriť, sme usúdili, že najvhodnejšie bude zjednotiť databázu členskej základne na jedno miesto, do jednej tabuľky, ktorá by sa postupne naplňala.

Každý člen dostane svoje poradové číslo, ktoré bude slúžiť pre jeho osobu ako jasný a nezameniteľný identifikátor. Tento identifikátor sa bude používať na konferenciách, rovnako aj ako VS (Variabilný Symbol) pre potreby platby členského alebo iné vnútorné potreby istej identifikácie.

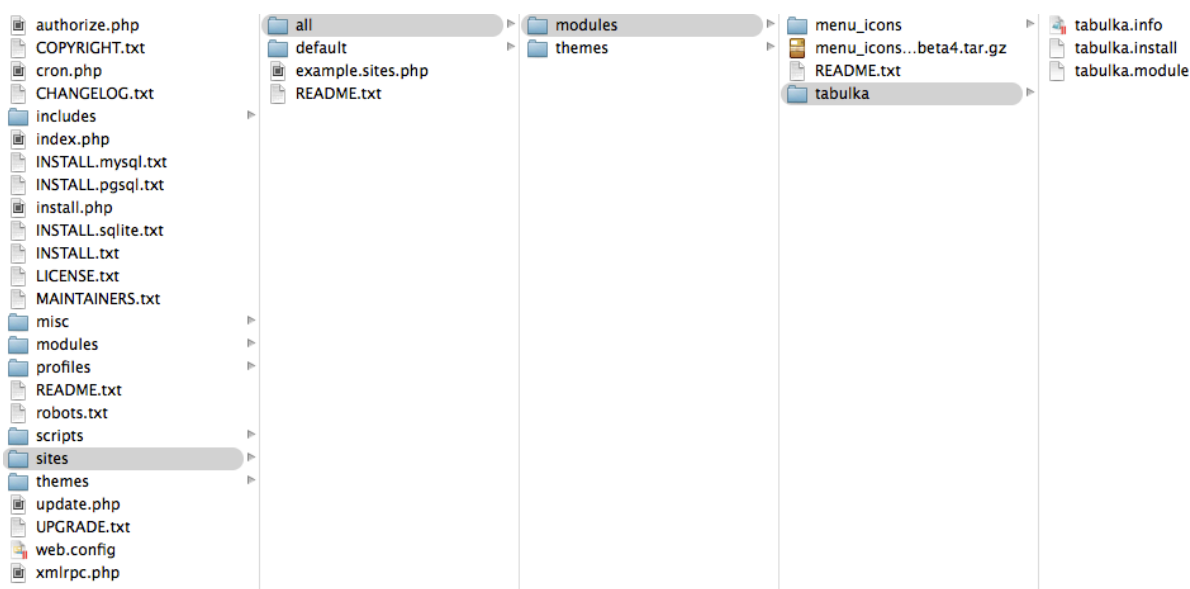
Štruktúra tabuľky, rozdelenie údajov v nej sa môže časom meniť. Presnú potrebu jednotlivých informácií v nej ukáže až prax. A preto nie je možné tvrdiť, že mnou navrhnutá štruktúra bude presne takto implementovaná stále na týchto stránkach. Avšak ja budem tou osobou, ktorá bude daný zoznam podľa pokynov objednávateľa (SSPCH) modifikovať.

Mnou vytvorený modul bude implementovaný do podstránky prístupnej pre administrátora, kde sa bude databáza postupne naplňovať. Takto získa vedenie spoločnosti prehľadný zoznam členov, ktorý budú môcť v prípade potreby ľahko exportovať (napr.: PDF,...).

3.1 Súbory modulu

Ako prvú vec čo si musíme ozrejmiť je, ako svoj nový modul nazveme. Treba si uvedomiť že od toho, aké mu dáme meno sa bude odvíjať aj jeho programovanie a neskoršie menenie názvu môže v určitých prípadoch vážne ohroziť stabilitu nášho modulu/systému alebo ho dokonca aj znefunkčniť. Preto je potreba si tento krok dostatočne premyslieť.

Nesmieme na začiatku názvu používať čísla, nepovolené znaky. Neodporúča sa ani interpunkcia kdekoľvek v jeho názve. Ďalej si treba opäť pripomenúť, že nami tvorené moduly je potreba ukladať do zložky *sites/all/modules* aby nám ich prípadný upgrade systému nevymazal a neprišli sme tak o našu drahocennú prácu. Pre lepšiu ilustráciu uvádzam obr. 2, ktorý presne dokumentuje kam uložiť súbory nášho vytvoreného modulu.



Obr. 2 Umiestnenie modulu a jeho štruktúra

Samotný modul, ako už je z predchádzajúceho obrázku zrejmé, musí mať v priečinku so svojim menom tri nevyhnutne (ak pracujeme s databázami) potrebné súbory. Týmito súbormi sú menomodulu.info, menomodulu.install (pri práci s databázami) a menomodulu.module. V prípade modulu, ktorý tvorím ja, som si vybral meno tabulka. Súbormi sú teda u mňa tabulka.info, tabulka.install a tabulka.module. Sú to tri základné súbory, ktoré musí obsahovať každý modul, ktorý ideme vytvárať. V prípade že sme skúsenejší programátor modulov pre Drupal, alebo potrebujeme vytvoriť komplexnejší modul, týchto súborov môže byť aj viac. Pre potreby môjho projektu zatiaľ postačia tieto základné.

Keď už vieme, akú skladbu súborov musí obsahovať každý z nami tvorených modulov, vysvetlím jednotlivé príkazy, ktoré sa v nich nachádzajú. Každý zo súborov je vysvetlený v jednej z podkapitol na nasledujúcich riadkoch.

3.1.1 Súbor tabulka.info

Súbor s názvom nášho modulu a koncovkou info nám slúži na identifikáciu nášho modulu, využíva sa najviac pri inštalácii. Kde nám podáva informácie o tom ako sa modul volá (u mňa „Tabuľka“). Vytvorí jeho jednoduchý popisok pre jednoduchú orientáciu o jeho základnom fungovaní, či potrebe prídavných modulov, ktoré pre svoju činnosť potrebuje.

Jeho presný zdrojový kód uvádzam v prílohe. Číslo verzie jadra systému Drupal, pre ktorý bol modul programovaný. A verziu nášho modulu (pre prípad, ak budeme v budúcnosti robiť opravu alebo korekcie v zdrojovom kóde).

Je vhodné venovať dostatočnú pozornosť týmto údajom, pretože ak zadáte niektorý údaj v nesprávnom tvare, môže spôsobiť nefunkčnosť. Ak napríklad zadám nesprávne verziu jadra pre ktorý je určený, spôsobí že sa nám nebude dať odkliknúť a inštalovať modul. Ak by sme nesprávne nastavili parameter description, nemusím náš modul identifikovať. Zdá sa to v tejto chvíli ako nepotrebné, ale ako je uvedené v obr. 2, do priečinka *sites/all/modules* neukladám iba moje moduly. Ukladám sem aj ďalšie moduly ktoré rozširujú možnosti nášho systému podľa našich požiadaviek a potrieb. V prípade, že tu bude väčšie množstvo modulov, môže byť hľadanie toho nášho niekedy komplikované.

Na nasledujúcich riadkoch sa pokúsim stručne popísať, čo ktorý riadok predstavuje:

<i>name</i> = “ <i>Tabuľka</i> ”	slúži ako názov – hlavný určujúci znak nášho modulu v rámci systému
<i>description</i> = “ <i>Vytvorí databázu členov.</i> ”	určuje aký bude popis modulu, prípadne aké ďalšie moduly k správne chodu potrebuje
<i>core</i> = “ <i>7.x</i> ”	udáva pre ktorú verziu Drupalu (jeho jadra) je určený
<i>version</i> = “ <i>7.x – 1.0</i> ”	informuje o tom, ktorá aktuálna verzia nášho modulu to je

Znenie zdrojového kódu uvádzam aj v prílohách k tejto práci ako prílohu A.

3.1.2 Súbor *tabulka.install*

Celý zdrojový kód je uvedený v prílohe B.

Definovanie niektorých dôležitých príkazov:

```
function tabulka_schema()    definuje nám príkazy, ktoré sa vykonajú pri samotnej inštalácii  
                               modulu do nášho redakčného systému  
  
$schema = array();           vytvorí sa pole  
  
$schema['tabulka'] = array() generuje pole s názvom tabulka
```


'description' => 'Tabuľka', pole bude mať index *Tabuľka*, bude sa nám zobrazovať, ak kurzorom prejdeme nad miesto v systéme, kde sa nám tabuľka vytvorila

```
'fields' => array(  
    'cisloclena' => array(  
        'description' => 'Primárny identifikátor mojej tabuľky - Číslo člena.',  
        'type' => 'serial',  
        'unsigned' => TRUE,  
        'not null' => TRUE,),
```

Tento zápis znamená že vo vytvorenej tabuľke je primárnym identifikátorom číslo člena. Teda žiadna hodnota v tomto stĺpci sa nemôže opakovať, musí byť jedinečná. Moja voľba hodnoty v tomto stĺpci bola, aby bola jedinečná. Pretože každý registrovaný člen dostane svoje poradové číslo, ktoré ho bude jasne a zreteľne identifikovať. Napríklad pri platbách ako VS. Dvaja členovia s tým istým číslom sú tak vylúčení a to i bez potreby si pamätať aké čísla už boli predchádzajúcim členom vydané. Typ *serial* nám v tabuľke udáva, že údaj na tejto pozícii musí byť typ *int()*.

```
'meno' => array(  
    'description' => 'Pole pre ukladanie mena člena.',  
    'type' => 'varchar',  
    'length' => 50,  
    'not null' => TRUE,  
    'default' => '',),
```

Príkaz do tabuľky pridá stĺpec pre meno člena. Ako vidíme, tento už nie je primárnym identifikátorom, preto sa hodnota v tomto prípade u jednotlivých členov smie zhodovať. Je logické, že niektorí členovia pravdepodobne budú mať rovnaké mená. Dĺžku údaju som obmedzil na maximálne 50 znakov. Túto hodnotu som zvolil preto, ak by sa našiel člen s dvomi, tromi menami. Čo nie je hlavne pri zahraničných nič nezvyklé.

```
'priezvisko' => array(  
    'description' => 'Pole pre ukladanie priezviska člena.',  
    'type' => 'varchar',  
    'length' => 50,  
    'not null' => TRUE,  
    'default' => '',),
```

Podobne ako predchádzajúci príkaz mám opäť v prípade priezviska obmedzenú maximálnu hodnotu na 50 znakov. Žiadna hodnota nie je vopred definovaná, treba ju zadať pri vkladaní člena do tabuľky.

```
'adresa' => array(  
    'description' => 'Pole pre ukladanie adresy člena.',  
    'type' => 'text',  
    'not null' => TRUE,)
```

V prípade tohto príkazu stojí za upozornenie typ text. To znamená pre systém, že môže očakávať text, aj čísla. Jeho dĺžka v tomto prípade nie je mnohou limitovaná, limitovaná je len systémom.

```
'telefon' => array(  
    'description' => 'Pole pre vloženie telefónneho čísla člena.',  
    'type' => 'int',  
    'unsigned' => TRUE,  
    'not null' => TRUE,)
```

Očakáva sa vloženie telefónneho čísla ako postupnosť čísel. Typ je int().

```
'email' => array(  
    'description' => 'Pole pre vloženie e-mailu člena.',  
    'type' => 'text',  
    'not null' => TRUE,)
```

Očakáva sa náplň tejto hodnoty vo forme textu.

```
'pracovisko' => array(  
    'description' => 'Pole pre vloženie pracoviska člena.',  
    'type' => 'text',  
    'not null' => TRUE,)
```

Záznam aj v tomto prípade bude typu text.

```
'primary key' => array('cisloclena'),
```

 primárny kľúč tabuľky je číslo člena

Chcel by som poukázať že pre žiaden stĺpec nesmiem definovať prázdnu hodnotu. Z toho vyplýva, že všetky polia pri napĺňaní tabuľky hodnotami musím mať korektne vyplnené. Zároveň sa ako oddeľovač používa čiarka.

Kód nám definuje že sa nám vytvorí tabuľka ako je zobrazená na obr. 3.

Projit Struktura SQL Vyhledávání Vložit Export Import Úpravy Vyprázdnit Odstranit									
	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra	Akce	
☐	cislociena	int(10)		UNSIGNED	Ne	Žádná	auto_increment		
☐	meno	varchar(50)	utf8_general_ci		Ne				
☐	priezvisko	varchar(50)	utf8_general_ci		Ne				
☐	adresa	text	utf8_general_ci		Ne	Žádná			
☐	telefon	int(10)		UNSIGNED	Ne	Žádná			
☐	email	text	utf8_general_ci		Ne	Žádná			
☐	pracovisko	text	utf8_general_ci		Ne	Žádná			
↑ Zaškrtnout vše / Odškrtnout vše Zaškrtnuté:									

Obr. 3 Vytvorená tabuľka

3.1.3 Súbor tabulka.module

Súbor tabulka.info bol informatívny, tabulka.install sa vykonával raz pri inštalácii. Príkazy, ktoré definujeme v tabulka.module, sa vykonávajú pri každom použití nášho modulu. Tu definujeme hlavnú náplň práce nášho modulu. Pre moje potreby bolo požadované, aby sa vytvorila v menu (blok Navigácia) položka Vkladanie záznamu. Ak na ňu klikneme kurzorom budeme presmerovaní na formulár (obr. 4) kde zadáme všetky potrebné údaje a v správnom tvare, dáme vložiť záznam a člen je do tabuľky pridaný v takom tvare, ako sme ho zadali do formulára.

Programovanie modulov pre Drupal je zvláštne tým, že využíva systém tzv. hook-ov, alebo po slovensky háčikov. Ktoré, ak ich človek používa tvoria jednu zo silných strán tohto systému.

Vysvetlivky k dôležitým častiam zdrojového kódu:

```
function tabulka_permission(){
  return array(
    'submit tabulka' => array(
      'title' => t('Submit tabulka'),
      'description' => t('Submit the tabulka form'),),
    'access tabulka submissions' => array(
      'title' => t('Access tabulka submissions'),
      'description' => t('Access the tabulka submissions'),),
  );}
```

Háčik `hook_permission()` definuje a vracia pole s nevyhnutnými informáciami o oprávneniach nášho modulu. Ako sú napríklad jednoduché textové reťazce, ktoré môžeme poslať funkcii `user_access()`, názov oprávnení čitateľný aj pre človeka a voliteľný popis. Predchádzajúce verzie vracali iba jednoduchý textový reťazec. [4]

```
function tabulka_menu(){
  $items = array();
  $items['form-example'] = array(
    'title' => 'Vkladanie záznamu',
    'type' => MENU_NORMAL_ITEM,
    'access arguments' => array('submit tabulka'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('tabulka_form',));
  $items['form-submissions'] = array(
    'title' => 'Výsledná tabuľka',
    'type' => MENU_NORMAL_ITEM,
    'access arguments' => array('access tabulka submissions'),
    'page callback' => 'tabulka_submissions',);
  $items['form-submissions'] = array(
    'title' => 'Zobraziť tabuľku',
    'type' => MENU_NORMAL_ITEM,
    'access arguments' => array('access tabulka submissions'),
    'page callback' => 'tabulka_submissions',);
  return $items;}
```

V tomto prípade nám `hook_menu` vytvorí položky menu v bloku.

'title' => Definuje nám text, ktorý sa objavuje v rôznych miestach nášho systému. Napríklad ako nadpis stránky, v elemente title, či ako podnadpis v administračnom rozhraní. [4]

'page callback' =>

'page arguments' => Definujeme nimi akciu, ktorá sa má vykonať ak užívateľ požaduje našu stránku. Page callback určuje, aká funkcia bude volaná, prvok page arguments obsahuje pole argumentov, ktoré dávame do tejto funkcie. Ak budeme vytvárať vlastnú funkciu, ktorá bude spracovávať, formátovať a vracat určité dáta, treba volať internú funkciu Drupalu `drupal_get_form()`. Táto vracia pole podľa

definovania rozhrania Form API. Argumentu predávame identifikátor, ktorý chceme zobrazit'. [4]

```
'access arguments' => array('submit tabulka'),
```

Tento riadok s prvkom access arguments prijíma pole ako reťazce textu s oprávneniami. Užívateľia, ktorí majú pridelené požadované oprávnenie, smú pristupovať k danému obsahu. Ostatným bude zobrazené oznámenie: **Prístup bol zamietnutý**. Definujeme oprávnenia pomocou hook_permission(). [4]

```
'type' => MENU_NORMAL_ITEM,
```

Type obsahuje bitovú masku, kde príznaky popisujú vlastnosti našich položiek, v súčasnosti existuje cez 20 konštánt pre položky menu. Základným typom je MENU_NORMAL_ITEM, ktorý nám zobrazuje položku v strome menu. [4]

```
function tabulka_form($form, &$amp;form_state){
```

```
    $form['cisloclena'] = array();
```

```
    $form['meno'] = array();
```

```
    $form['priezvisko'] = array();
```

```
    $form['adresa'] = array();
```

```
    $form['telefon'] = array();
```

```
    $form['email'] = array();
```

```
    $form['pracovisko'] = array();
```

```
    $form['submit'] = array();
```

```
    return $form;}
```

Funkcia spätného volania stránky metódou drupal_get_form(). V rozhraní Form API vracia štrukturované pole reprezentujúce formulár v HTML. Drupal následne z tohto kódu generuje formulár a zobrazuje ho užívateľovi. Funkcia drupal_get_form() prijíma argument identifikátor formulára. Tento identifikátor môže byť v zásade ľubovoľný, ale musí byť jedinečný v Drupal-e. Preto najčastejšie volíme identifikátor tvaru <meno_modulu>_<popis>-form. [4]

```
function tabulka_form_validate($form, &$amp;form_state){
```

```
    if (!is_numeric($form_state['values']['cisloclena'])) {
```

```
        form_set_error('cisloclena', t('Treba vložiť správne číslo člena!!!'));
    }
```

```
    return FALSE;}
```

```
    return TRUE;}
```

Táto funkcia mi kontroluje aby bolo zadané správne číslo člena. Číslo člena je primárny identifikátor.

```
function tabulka_form_submit($form, &$form_state){
    $cisloclena = db_insert('tabulka')
    ->fields(array(
        'cisloclena' => $form_state['values']['cisloclena'],
        'meno' => $form_state['values']['meno'],
        'priezvisko' => $form_state['values']['priezvisko'],
        'adresa' => $form_state['values']['adresa'],
        'telefon' => $form_state['values']['telefon'],
        'email' => $form_state['values']['email'],
        'pracovisko' => $form_state['values']['pracovisko'],))
    ->execute();
    drupal_set_message(t('Položka formuláru bola pridaná. '));}

```

Funkcia nám pridáva záznam o užívateľovi (riadok) do našej vytvorenej tabuľky.

Výstupom z tohto modulu bude formulár pre pridanie člena do databázy, ktorý je vidno na obr. 4.

Obr. 4 Formulár

```
function tabulka_submissions(){
    $query = db_select('tabulka','fe');
    $query
    ->fields('fe', array('cisloclena','meno', 'priezvisko', 'adresa', 'telefon', 'email',
    'pracovisko'))

```

```

->range(0, 50)
->orderBy('fe.cisloclena');
$results = $query->execute();
$header = array(t('Číslo'), t('Meno'), t('Priezvisko'), t('Adresa'), t('Telefónne číslo'),
t('E-mail'), t('Pracovisko'));
$rows = array();
foreach($results AS $result){
    $rows[] = array(
        check_plain($result->cisloclena),
        check_plain($result->meno),
        check_markup($result->priezvisko),
        check_markup($result->adresa),
        check_markup($result->telefon),
        check_markup($result->email),
        check_markup($result->pracovisko),);}
    return theme('table', array('header' => $header, 'rows' => $rows));}

```

Tento kód nám vracia záznamy, ktoré sú uložené v našej tabuľke, čo je vlastne databáza členov. Zobrazí ju tak, ako je zobrazené na obr. 5.

Domov

Navigácia

- Moja stránka
- Pridať obsah
- Vkladanie záznamu
- Zobrazíť tabuľku

Zobrazíť tabuľku

Číslo	Meno	Priezvisko	Adresa	Telefónne číslo	E-mail	Pracovisko
13	Samuel	Sčitaný	Radlinského, 9, 81237, Bratislava, SK	252923011	scitany.samuel@stuba.sk	FCHPT STU

Obr. 5 Výsledná tabuľka

3.2 Budúcnosť modulu

Funkčnosť modulu bola otestovaná na vzorových údajoch. Po týchto skúškach som dospel k záveru, že modul splnil požiadavky pre ktoré bol vytvorený.

Jeho budúcnosť by mohla byť v tom, aby sa vytvorila možnosť editácie jednotlivých členov SSPCH. Bez nutnosti pracovať s phpMyAdmin-om (v našom prípade), alebo iným nástrojom pre správu databáz. Zároveň môže byť ľahko zmenený počet stĺpcov (pridávaním kódu pred inštaláciou). Údaje, ktoré sú potrebné, a ktoré naopak nadbytočné, ukáže až prax.

Nie je v pláne, že by sa mnou vytvorený modul dostal na stránky Drupal-u. Z časti preto, že je navrhnutý na jednu konkrétnu potrebu konkrétnej spoločnosti a aj preto že jej zatiaľ chýba možnosť bližšej administrácie, aby sa dala prispôbovať požiadavkám väčšieho množstva tvorcov internetových stránok.

Nesmieme na začiatku názvu používať čísla, nepovolené znaky. Neodporúča sa ani interpunkcia kdekoľvek v jeho názve. Ďalej si treba opäť pripomenúť, že nami tvorené moduly je potreba ukladať do zložky *sites/all/modules* aby nám ich prípadný upgrade systému nevymazal a neprišli sme tak o našu drahocennú prácu. Pre lepšiu ilustráciu uvádzam obr. 2, ktorý presne dokumentuje, kam uložiť súbory nášho vytvoreného modulu.

4 Budúcnosť stránky SSPCH

Predpokladom je v blízkej budúcnosti nájsť vhodný webový server, kde by bolo možné nami vytvorenú internetovú stránku umiestniť a aj s vytvoreným modulom ju zaviesť do ostrej prevádzky. Doplniť možnosťou priameho vytvorenia PDF zo stránky a možnosti zdieľať jej obsah aj cez iné sociálne siete ako je Facebook.

Po naplnení databázy, pridaní užívateľských profilov bude členom doručené ich prihlasovacie meno a heslo do systému.

Zároveň s nimi by bolo vhodné vytvoriť PDF dokumenty s jednoduchými návodmi, ako sa na stránke pohybovať, aké sú jej jednotlivé funkcie. Dokumenty by sa mali zároveň s novými podstatnými funkciami v prípade potreby tiež aktualizovať.

5 Záver

Pri programovaní modulu si bolo treba naštudovať značné množstvo literatúry, hlavne presných spôsobov zápisu zdrojového kódu pri jednotlivých príkazoch a ich modifikáciou pre potreby modulu v systéme Drupal 7.

Prvým cieľom bolo vytvoriť funkčnú a užívateľsky príjemnú internetovú stránku Slovenskej spoločnosti priemyselnej chémie (SSPCH), ktorá bola vytvorená a bude v budúcnosti v nových verziách doplnená o ďalšie možnosti.

Druhým cieľom bolo navrhnuť štruktúru a programovo realizovať modul pre tvorbu databázy členskej základne. Modul je plne funkčný a implementovaný do systému Drupal, ktorý bude na stránke spoločnosti prístupný administrátorom.

V práci som načrtol základy potrebné pre tvorbu modulov v tomto systéme, ktoré sa dokáže človek naučiť relatívne rýchlo. Na internete sa nachádza veľké množstvo stránok s video návodmi a komentárom.

Pri väčších projektoch (moduly a témy) je nutné mať hlbšie znalosti programovania, ktoré sa dajú získať predovšetkým praxou a ustavičným zlepšovaním sa v práci s týmto systémom.

Zoznam použitej literatúry

1. <<http://drupal.blog.zive.cz/2008/08/historia-drupalu/>> Dostupné na internete 20. 5. 2014
2. <<http://sk.wikipedia.org/wiki/Drupal>> Dostupné na internete 20. 5. 2014
3. <<https://drupal.org/>> Dostupné na internete 20. 5. 2014
4. BUTCHER, M. a i., *Drupal Programování modulů*, 1. vyd. Brno, Computer Press, a.s., s. 124-126, 209, ISBN 978-80-251-3440-5

Prílohy

- Príloha A: Zdrojový kód súboru tabulka.info
- Príloha B: Zdrojový kód súboru tabulka.install
- Príloha C: Zdrojový kód súboru tabulka.module
- Príloha D: CD médium – práca v elektronickej podobe, so súbormy modulu.

Príloha A: Zdrojový kód súboru

tabulka.info

```
name = "Tabuľka"
```

```
description = "Vytvorí databázu členov."
```

```
core = "7.x"
```

```
version = "7.x-1.0"
```

Príloha B: Zdrojový kód súboru

tabulka.install

```
<?php
/**
 * Implements hook_schema().
 */
function tabulka_schema() {
  $schema = array();
  $schema['tabulka'] = array(
    'description' => 'Tabuľka',
    'fields' => array(
      'cisloclena' => array(
        'description' => 'Primárny identifikátor mojej tabuľky - Číslo člena.',
        'type' => 'serial',
        'unsigned' => TRUE,
        'not null' => TRUE,
      ),
      'meno' => array(
        'description' => 'Pole pre ukladanie mena člena.',
        'type' => 'varchar',
        'length' => 50,
        'not null' => TRUE,
        'default' => "",
      ),
      'priezvisko' => array(
        'description' => 'Pole pre ukladanie priezviska člena.',
        'type' => 'varchar',
        'length' => 50,
        'not null' => TRUE,
        'default' => "",
      ),
    ),
  );
}
```

```

'adresa' => array(
    'description' => 'Pole pre ukladanie adresy člena.',
    'type' => 'text',
    'not null' => TRUE,
),
'telefon' => array(
    'description' => 'Pole pre vloženie telefónneho čísla člena.',
    'type' => 'int',
    'unsigned' => TRUE,
    'not null' => TRUE,
),
'email' => array(
    'description' => 'Pole pre vloženie e-mailu člena.',
    'type' => 'text',
    'not null' => TRUE,
),
'pracovisko' => array(
    'description' => 'Pole pre vloženie pracoviska člena.',
    'type' => 'text',
    'not null' => TRUE,
),
),
'primary key' => array('cisloclena'),
);
return $schema;
}

```

Príloha C: Zdrojový kód súboru

tabulka.module

```
<?php
/**
 * Implements hook_permission().
 */
function tabulka_permission(){
  return array(
    'submit tabulka' => array(
      'title' => t('Submit tabuľka'),
      'description' => t('Submit the tabulka form'),
    ),
    'access tabulka submissions' => array(
      'title' => t('Access tabulka submissions'),
      'description' => t('Access the tabulka submissions'),
    ),
  );
}
/**
 * Implements hook_menu().
 */
function tabulka_menu(){
  $items = array();
  $items['form-example'] = array(
    'title' => 'Vkladanie záznamu',
    'type' => MENU_NORMAL_ITEM,
    'access arguments' => array('submit tabulka'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('tabulka_form'),
  );
  $items['form-submissions'] = array(
```



```

        'title' => 'Výsledná tabuľka',
        'type' => MENU_NORMAL_ITEM,
        'access arguments' => array('access tabulka submissions'),
        'page callback' => 'tabulka_submissions',
    );
    $items['form-submissions'] = array(
        'title' => 'Zobraziť tabuľku',
        'type' => MENU_NORMAL_ITEM,
        'access arguments' => array('access tabulka submissions'),
        'page callback' => 'tabulka_submissions',
    );
    return $items;
}
/**
 * My exemple form.
 */
function tabulka_form($form, &$form_state){
    $form['cisloclena'] = array(
        '#type' => 'textfield',
        '#title' => t('Číslo člena:'),
        '#size' => 10,
        '#maxlength' => 10,
        '#required' => TRUE,
        '#description' => t('Prosím vložte číslo člena v správnom formáte!!!'),
    );
    $form['meno'] = array(
        '#type' => 'textfield',
        '#title' => t('Meno:'),
        '#size' => 20,
        '#maxlength' => 128,
        '#required' => TRUE,
    );
    $form['priezvisko'] = array(
        '#type' => 'textfield',

```

```

        '#title' => t('Priezvisko:'),
        '#size' => 30,
        '#maxlength' => 128,
        '#required' => TRUE,
    );
    $form['adresa'] = array(
        '#title' => t('Adresa:'),
        '#type' => 'textarea',
        '#required' => TRUE,
        '#description' => t('V nasledujúcom formáte: Ulica, popisné číslo, PSČ, Mesto,
Krajina.'),
    );
    $form['telefon'] = array(
        '#type' => 'textfield',
        '#title' => t('Telefónne číslo:'),
        '#size' => 15,
        '#maxlength' => 15,
        '#required' => TRUE,
        '#description' => t('Prosím vložte telefónne číslo bez predvoľby krajiny.'),
    );
    $form['email'] = array(
        '#title' => t('E-mail:'),
        '#type' => 'textfield',
        '#description' => t('Sem vložte e-mailovu adresu člena'),
        '#required' => TRUE,
    );
    $form['pracovisko'] = array(
        '#title' => t('Pracovisko:'),
        '#type' => 'textfield',
        '#description' => t('Sem vložte pracovisko člena'),
        '#required' => TRUE,
    );
    $form['submit'] = array(
        '#type' => 'submit',

```

```

        '#value' => t('Vložit záznam'),
    );
    return $form;
}
/**
 * Validation handler for the form_example_form.
 */
function tabulka_form_validate($form, &$form_state){
    if (!is_numeric($form_state['values']['cisloclena'])){
        form_set_error('cisloclena', t('Treba vložit správne číslo člena!!!'));
        return FALSE;
    }
    return TRUE;
}
/**
 * Submit handler for the form_example_form. Inserts entry into the database.
 */
function tabulka_form_submit($form, &$form_state){
    $cisloclena = db_insert('tabulka')
    ->fields(array(
        'cisloclena' => $form_state['values']['cisloclena'],
        'meno' => $form_state['values']['meno'],
        'priezvisko' => $form_state['values']['priezvisko'],
        'adresa' => $form_state['values']['adresa'],
        'telefon' => $form_state['values']['telefon'],
        'email' => $form_state['values']['email'],
        'pracovisko' => $form_state['values']['pracovisko'],
    ))
    ->execute();
    drupal_set_message(t('Položka formuláru bola pridaná.'));
}
/**
 * View the form submissions.
 */

```

```

function tabulka_submissions(){
$query = db_select('tabulka','fe');
$query
    ->fields('fe', array('cisloclena', 'meno', 'priezvisko', 'adresa', 'telefon', 'email',
'pracovisko'))
    ->range(0, 50)
    ->orderBy('fe.cisloclena');
$results = $query->execute();
$header = array(t('Číslo'), t('Meno'), t('Priezvisko'), t('Adresa'), t('Telefónne číslo'),
t('E-mail'), t('Pracovisko'));
$rows = array();
foreach($results AS $result){
    $rows[] = array(
        check_plain($result->cisloclena),
        check_plain($result->meno),
        check_markup($result->priezvisko),
        check_markup($result->adresa),
        check_markup($result->telefon),
        check_markup($result->email),
        check_markup($result->pracovisko),
    );
}
return theme('table', array('header' => $header, 'rows' => $rows));
}

```