

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Čebašek

INTEGRACIJA SPLETNEGA VMESNIKA CMS V SPLETNO STRAN

DIPLOMSKO DELO

VISOKOŠOLSKE STROKOVNE ŠTUDIJSKE PROGRAMA PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2011



Št. naloge: 00082/2011

Datum: 01.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JURE ČEBAŠEK**

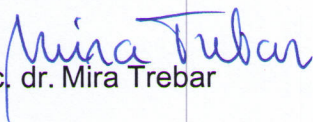
Naslov: **INTEGRACIJA SPLETNEGA VMESNIKA CMS V SPLETNO STRAN**
INTEGRATION OF THE CMS WEB INTERFACE TO A WEB PAGE

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje


Tematika naloge:

V diplomskem delu naj kandidat analizira možnosti urejanja spletne strani za uporabnike, ki nimajo znanj o spletnem programiranju, jezikih in podatkovnih bazah. V ta namen naj predstavi spletni vmesnik CMS s možnostmi, ki jih tovrstni sistemi omogočajo tako da se osredotoči na urejanje spletne strani z vidika uporabnika. S predstavljenim sistemom naj izdela spletno stran za naročnika, ki želi promovirati prodajo mobilnih telefonov in izpostavi prednosti in slabosti tovrstnih sistemov.

Mentor:


doc. dr. Mira Trebar

Dekan:


prof. dr. Nikolaj Zimic



Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani

Jure Čebašek, z vpisno številko 63050299,

sem avtor diplomskega dela z naslovom:

Integracija spletnega vmesnika CMS v spletno stran

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 3. julij 2011

Podpis avtorja:

Povzetek

Cilj diplomske naloge je integracija spletnega vmesnika CMS v obstoječo statično HTML stran, ki uporabnikom omogoča urejanje spletne strani brez znanja o spletnih jezikih in podatkovnih bazah. V delu najprej predstavimo spletni vmesnik CMS in osnovne pojme ter funkcije, ki jih tovrstni sistemi omogočajo. Sledi opis uporabljenih tehnologij, tehnik in orodij, s pomočjo katerih smo prišli do želene rešitve danega problema.

Jedro diplomske naloge je razvoj spletne strani s spletnim vmesnikom CMS. V prvem delu razvoja se ukvarjamo s kreiranjem podatkovne baze, ter prenosom datotek, ki so potrebne za delovanje in povezavo podatkovne baze in spletnega vmesnika CMS. V drugem delu razvoja se osredotočimo na tipizacijo strani in podstrani v spletnem vmesniku CMS. V tretjem delu razvoja se ukvarjamo predvsem s tehnologijami kot so: HTML, PHP, CSS, MySQL in vključitev spletnega vmesnika CMS v statično spletno stran.

Po razvoju sledi še podrobna predstavitev spletnega vmesnika CMS za dodajanje vsebine na spletne strani z osnovnimi znanji programov Microsoft Word ali Open Office Write. Aplikacija omogoča uporabniku dodajanje in urejanje strani, dodajanje slik in galerij na spletno stran. V tem delu se osredotočimo na praktični primer urejanja spletne strani za prodajo mobilnih telefonov z vidika uporabnika, ki ima na voljo spletni vmesnik CMS.

V zaključku objektivno izpostavimo prednosti in slabosti tovrstnih sistemov ter podamo splošne ugotovitve in ideje, do katerih smo prišli pri pisanju diplomske naloge.

Ključne besede: spletna stran, spletni vmesnik, CMS – upravljanje vsebin

Abstract

The aim of the thesis is the integration of CMS web interface into an existing static HTML page that allows users to edit websites without knowledge of web languages and databases.

At first, we present the CMS web interface, basic concepts and functions that enable such systems. We continue with a description of technologies, techniques and tools to achieve the desired solution of a given problem.

The core of the thesis is the development of a web page using a CMS web interface. In the first part, we are dealing with creation of the database, and transfer files that are necessary for operations and how to link a database and CMS web interface. In the second part of the development, the focus is primarily on web pages and sub-types that are used in the CMS web interface. In the third part, we are dealing with technologies such as HTML, PHP, CSS, MySQL, and the integration of the CMS web interface to a static Web page.

At the end, a description of how to add content to websites with basic knowledge of MS Word (Microsoft Word) or OO Write (OpenOffice Write) is presented. Application allows the user to add and edit pages, add images and galleries on the site. Mainly, the web site design is explained from user's perspective and presents the arrangement of a web site with CMS web interface.

In conclusion, strengths and weaknesses for such a system are objectively highlighted. The important findings with ideas, to which we came during the work, are also discussed.

Keywords: webpage, web interface, CMS – Content Management System

Kazalo

1	UVOD	1
2	SISTEMI ZA UPRAVLJANJE SPLETNIH VSEBIN	3
2.1	Potreba po ažurnosti informacij	3
2.2	Prednosti CMS rešitev	3
2.3	Slabosti in težave CMS rešitev	5
2.4	Opis CMS sistema	6
2.4.1	E – novice.....	6
2.4.2	Raziskovalec.....	7
2.4.3	Formular.....	7
2.4.4	Zavihek uporabniki	8
2.4.5	Zavihek nastavitve.....	9
2.4.6	Zavihek mCMS.....	9
2.4.7	Zavihek formularji.....	10
3	TEHNOLOGIJE IN ORODJA	11
3.1	Predstavitev tehnologij.....	11
3.1.1	Jezik HTML.....	11
3.1.2	Jezik PHP.....	11
3.1.3	Jezik JavaScript	11
3.1.4	Kaskadne stilske predloge CSS.....	12
3.1.5	MySQL.....	13
3.2	Predstavitev orodij	13
3.2.1	Orodje PhpMyAdmin.....	13
3.2.2	TextMate.....	13
3.2.3	XAMPP	14
3.2.4	Urejevalnik FCKeditor	14
3.2.5	Spletni brskalnik Mozilla Firefox.....	14
3.2.6	Firebug	15
4	ZAHTEVE NAROČNIKA.....	17

4.1	Izgled spletne strani	17
4.1.1	Vstopna stran	17
4.1.2	Podstran s tremi stolpci	18
4.1.3	Podstran z dvema stolpcema	18
4.1.4	Podstran z galerijo	18
4.1.5	Podstran kontakti.....	18
4.2	Funkcionalne zahteve	19
5	INTEGRACIJA CMS SISTEMA V STATIČNO SPLETNO STRAN	21
5.1	Izbira statične HTML predloge	22
5.2	Kreiranje podatkovne baze.....	23
5.2.1	Kreiranje baze.....	23
5.2.2	Kreiranje uporabnika podatkovne baze.....	23
5.2.3	Kreiranje tabel v podatkovni bazi	24
5.2.3.1	Stran	25
5.2.3.2	Tip.....	26
5.2.3.3	Galerija	27
5.2.3.4	Atribut	28
5.3	Prenos datotek na strežnik.....	29
5.4	Spreminjanje datotek za dostop do podatkovne baze	30
5.5	Kreiranje tipov strani v CMS sistemu.....	30
5.5.1	Kreiranje tipa strani Vstopna stran.....	31
5.5.2	Kreiranje tipa strani Podstran	34
5.5.3	Kreiranje tipa strani Kontakt	35
5.5.4	Kreiranje tipa strani Izjave naših strank.....	35
5.6	Kreiranje strukture strani v CMS sistemu.....	35
5.7	Izvedba integracije CMS sistema v PHP različici.....	38
5.7.1	Razlaga kode PHP kreiranih funkcij iz dodatka C	39
5.7.2	Razlaga kode PHP določitve tipov strani iz dodatka B.....	40
5.7.2.1	Določitev tipov podstrani v PHP kodi.....	40

5.7.2.2	Integracija menija v kodi PHP na spletni strani.....	41
5.7.2.3	Integracija v kodi PHP podstran tipa podstran z galerijo.....	42
5.8	UPORABA CMS SISTEMA.....	43
5.8.1	Urejanje vsebine z FCK editorjem.....	43
5.8.2	Dodajanje slik v galerijo.....	44
6	SKLEP.....	45
7	VIRI IN LITERATURA.....	46
	DODATEK A.....	47
	DODATEK B.....	50
	DODATEK C.....	57
	DODATEK D.....	60

Seznam kratic

<u>kratica</u>	<u>angleški izraz</u>	<u>slovenski izraz</u>
CMS	Content Management System	sistem za upravljanje vsebin
CSS	Cascading Style Sheet	kaskadni stilski slogi
FAQ	Frequently Asked Questions	pogosto zastavljena vprašanja
HTML	HyperText Markup Language	označevalni jezik
IIS	Internet Information Server	informacijski strežnik
MAC OS	Macintosh Operating System	operacijski sistem Macintosh
SQL	Structured Query Language	strukturni poizvedben jezik
WYSIWYG	What You See Is What You Get	to kar vidiš to dobiš
XML	Extensible Markup Language	razširljiv označevalni jezik

1 UVOD

Na tržišču se je v zadnjih letih pojavilo veliko število odprtokodnih, pa tudi komercialnih rešitev za upravljanje spletnih vsebin. Vsak sistem ima svoj nabor funkcionalnosti, ki omogočajo upravljanje sistema in ravno s funkcionalnostmi želijo prepričati uporabnike, da se odločijo prav za njihov sistem. Ker večina uporabnikov, veliko pričakuje od sistema, za katerega se bodo odločili, se pogosto prehitro odločijo za napačen sistem za upravljanje spletne vsebine, oziroma ugotovijo, da je administracija takšnega sistema prezahtevna, ker vključuje veliko funkcij, ki jih večina uporabnikov niti ne potrebuje. Z vključevanjem prevelikega števila funkcij povzročajo zmedo med uporabniki, sistem pa postane nepregleden.

Večina rešitev sistemov za upravljanje spletne vsebine deluje na principu predlog spletnih strani, kar uporabnike omejuje pri izbiri predloge za njihovo spletno stran ali pa zvišuje stroške prenove spletne strani. Večina sistemov ne omogoča integracije spletnega vmesnika v statično spletno stran, napisano v označevalnem jeziku ali pa je integracija preveč kompleksna za navadne uporabnike [2].

V diplomskem delu bomo predstavili integracijo CMS vmesnika v statično spletno stran napisano v označevalnem jeziku HTML. Opisali bomo vse postopke, ki so potrebni za tovrstno integracijo sistema in prikazali izdelavo dinamične spletne strani s tovrstnim sistemom.

2 SISTEMI ZA UPRAVLJANJE SPLETNIH VSEBIN

Sistem za upravljanje spletnih vsebin (ang. Content Management System – CMS) lahko razumemo kot skupek programskih orodij, ki omogočajo učinkovito in hitro izdelavo in vzdrževanje spletnih strani z uporabo CMS sistema. Nudijo učinkovito uporabo pravil, procesov in delovnih tokov. Cilj je zagotoviti centralizirano upravljanje za skrbnike spletnih strani in decentralizirano upravljanje za urednike, ki kreirajo, urejajo in objavljajo vsebino. Sistemi za upravljanje z vsebino so tako nepogrešljivi pri celotnem življenjskem ciklu organizacijsko pomembnih vsebin [1].

2.1 Potreba po ažurnosti informacij

Podjetje, ki za ažuriranje spletne vsebine najame zunanje izvajalce, se lahko poleg višjih stroškov sooči tudi s problemi, ki so povezani z odzivnim časom vzdrževalcev vsebine. Mnoge uporabnike moti, da pogostokrat mine preveč časa od posredovanja zahtevka za spremembo vsebine do njegove realizacije. Za marsikaterega uporabnika, ki je prepuščen na milost in nemilost zunanjih vzdrževalcev vsebin, je lahko takšen odnos poguben. Neažurna vsebina povzroči poleg odliva obiskovalcev s spletne strani tudi veliko poslovno škodo.

V primeru, ko se srečujejo s takšnimi in podobnimi težavami, bi še kako prav prišla programska rešitev, ki bi jim omogočala, da bi brez znanja spletnega programiranja sami upravljali tako z vsebino, kakor tudi s celotno spletno stranjo. Glede na to, da imajo vsa podjetja potrebo po zmanjševanju stroškov na vseh ravneh poslovanja in po čimbolj enostavni objavi ažurnih informacij na spletni strani, se zdijo sistemi za upravljanje z vsebinami ena izmed najbolj optimalnih rešitev za zagotavljanje omenjenih potreb.

Sistemi CMS so namenjeni samostojnemu ustvarjanju, urejanju in vzdrževanju spletnih vsebin ter upravljanju s celotno spletno stranjo, zato se lahko podjetja z njihovo pomočjo delno ali v celoti izognejo stroškom vzdrževalnih pogodb ali stroškom povezanih z najemom programerjev, ki skrbijo za ažuriranje vsebine.

2.2 Prednosti CMS rešitev

Zaradi vedno večjega povpraševanja po programskih rešitvah, ki odpravljajo omenjene probleme, beležimo v zadnjih letih pravo ekspanzijo komercialnih (plačljivih) in odprtokodnih (brezplačnih) sistemov za upravljanje s spletnimi vsebinami. Čeprav so sistemi CMS namenjeni različnim tipom spletnih strani in pokrivajo zelo raznolik spekter funkcionalnosti, lahko rečemo, da imajo vsaj v osnovi enoten skupen cilj: na enostaven način (brez programiranja in urejanja HTML kode) omogočiti objavo vsebin na spletni strani, ne da bi imel uporabnik pri tem težave z načinom in obliko prikaza.

Uporabnost sistemov CMS je v tem, da omogočajo preprosto in celovito obvladovanje spletnega mesta tudi tehnično nepodkovanim uporabnikom. Spreminjanje vsebine na spletni strani namreč ni pogojeno s poznavanjem programskih jezikov in spletnih tehnologij. Urejanje spletnih vsebin s pomočjo sistema CMS je resnično zelo enostavno opravilo, ki je zelo podobno uporabi enega izmed urejevalnikov besedil kot sta npr. Word ali Notepad. Poleg enostavne uporabe je ena izmed poglobitvenih prednosti uporabe sistema CMS tudi ta, da lahko uporabnik prek spletnega brskalnika kadarkoli in od kjerkoli ureja vsebino spletnega mesta.

Ne glede na vrsto spletne prisotnosti (predstavitvena stran, trgovina, portal, itd.) se lahko z uporabo ustreznega sistema CMS vse spremembe na spletni strani opravi hitro, enostavno in predvsem takrat, ko uporabnik to želi. V skrb za ažurnost spletne predstavitve so lahko vključeni tudi sodelavci. Dostop do sistema CMS ima lahko več oseb znotraj podjetja, kar pomeni, da se opravilo vzdrževanja spletnih strani porazdeli med točno določene osebe (npr. prodajni referent spreminja cenike, kadrovnik skrbi za ažurnost podatkov o zaposlenih, tržnik za akcijsko ponudbo, ipd.). Z uporabo CMS sistema lahko podjetje svoje obiskovalce redno seznanja s svežimi in aktualnimi informacijami, z novostmi iz prodajnega programa, z akcijskimi ponudbami, novimi članki, nasveti, zanimivostmi, itd.

Poleg urejanja tekstov, kjer ima uporabnik možnost upravljanja z vrsto, velikostjo, barvo in stilom pisave, lahko tudi ureja in spreminja poravnavo besedila, dodaja, briše in ureja slike, tabele, grafe, izdelke v spletni trgovini, nadpovezave, fotografije in podobno. Pri urejanju vsebine je za uporabnike zelo pomembno, da je videz vsebinskih posodobitev, ki jih opravi znotraj sistema CMS, enak končnemu videzu na spletni strani (princip WYSIWYG – »What you see is what you get«).

Sistemi CMS ne omogočajo le urejanja vsebine, temveč tudi celovito upravljanje spletnega mesta, vključno z njegovo strukturo (navigacijo), oblikovno podobo, pravicami uporabnikov in z vsemi moduli, ki so vključeni v spletno mesto (ankete, forum, novice, ipd). Tako lahko npr. uporabniki popravijo in dodajajo novice, ki se prikazujejo na osrednji strani njihove spletne strani, kar v urejevalniku, ki se nahaja v sistemu, ne da bi se ukvarjali z oblikovanjem novice v HTML kodi ali z njenim prenosom na spletni strežnik [3].

Kljub vsem funkcionalnostim, ki jih omogočajo sistemi CMS, se je potrebno zavedati, da ni vsak primeren za vsak tip spletne strani. Nekateri sistemi puščajo tako malo kreativne svobode, da se lahko že takoj pri vstopu na spletno stran ugotovi s katerim sistemom je bila narejena. To je seveda posledica uporabe že vnaprej pripravljenih predlog, istega tipa navigacije ter dodatnih modulov, ki delujejo po istem principu in imajo zelo podoben izgled. Ko že ravno omenjamo izgled spletne strani, ki je narejena z uporabo sistema CMS, velja omeniti še nekaj zelo pomembnega. Enostavnost rokovanja, možnost spreminjanja in nastavljanja oblikovnih predlog so zaželene lastnosti vsakega sistema CMS, vendar

izdelovalci spletnih strani še posebej opozarjajo uporabnike na nevarnost, do katere lahko pride, ko po zaključku projekta prepustijo izdelano spletno stran, v samostojno upravljanje naročniku. Le-ti lahko namreč pri uporabi CMS orodja pretiravajo z uporabo orodnih vrstic, ki omogočajo oblikovne popravke. Tako se kaj hitro zgodi, da uporabniki zaradi svojih lastnih kreativnih podvigov, ki s pravo estetiko nimajo nič skupnega, popolnoma zmaličijo oblikovno podobo spletnih strani. Tisti, ki imajo dovoljenje za spreminjanje vsebine še posebej radi pretiravajo z velikostjo in poudarjanjem pisave ter z uporabo večjega števila barvnih odtenkov pisave [1].

Dodatna prednost sistema za upravljanje spletnih vsebin je tudi zmanjšanje stroškov. Uporabnikom oziroma lastnikom spletnih strani, ki nimajo znanja o pisanju HTML ali PHP kode ni potrebno za vsako novo objavo plačevati ponudnikom tovrstnih storitev, ampak lahko spremembe objavijo sami. Uporaba tovrstnih sistemov je močno podobna pisanju dokumentov v programih kot so MS Word (Microsoft Word) ali OO Write (OpenOffice Write). Proces kreiranja spletnih strani in podstrani se poenostavi ter zmanjša obseg vzdrževanja, ker ni komunikacije med naročnikom sprememb in izvajalcem. Naročnik vse spremembe objavi sam.

2.3 Slabosti in težave CMS rešitev

Vsak uporabnik na prednosti in slabosti, ki jih imajo CMS rešitve, gleda drugače. Kako tudi ne, ko pa ima vsak lastnik spletne strani svoje specifične potrebe in želje, ki jih mora izpolnjevati sistem CMS. Kar je prednost za nekoga, je lahko slabost za nekoga drugega in obratno. Težave s katerimi se soočajo uporabniki CMS sistemov lahko nastanejo izključno kot posledica izbire, med komercialno in odprtokodno verzijo sistema. Poleg nekaterih osnovnih prednosti, ki jih uporabnikom prinašajo komercialni sistemi CMS (ponudnik nudi podporo uporabnikom, omogoča nadaljnji razvoj sistema, sistem lahko proti doplačilu prilagodi specifičnim željam uporabnikov), je lahko njihova uporaba povezana tudi z določenimi slabostmi. Kot prvo velja omeniti stroške nakupa programske opreme in nadaljnjega vzdrževanja, ki se jim lahko z uporabo odprtokodne rešitve izognemo. Dodatne stroške lahko povzročijo nove funkcionalnosti, ki jih želimo preko sistema vključiti na spletno stran ter dodatne prilagoditve samega sistema.

Tudi odprtokodni sistemi CMS imajo nekatere slabosti. Uporabniki se lahko soočijo s precejšnjimi težavami, če se razvoj sistema preneha ali če zaide v slepo ulico, kar pri brezplačnih različicah sploh ni redek pojav. Ves razvoj sistema se odvija na prostovoljni bazi, zato razvijalci ne dajo nobenih garancij za pravilno delovanje sistema kot tudi ne garancij, da bodo skrbeli za popravke in nadgradnje sistema ter ne zagotavljajo pomoči uporabnikom. Res pa je, da nam zaradi odprte kode lahko nudi pomoč in opravlja nadgradnje skoraj vsak programer, ki obvladuje programski jezik PHP [2].

2.4 Opis CMS sistema

Kot najbolj poznani in uporabljeni CMS sistemi, tudi naš temelji na programskem jeziku PHP v povezavi z MySQL podatkovno bazo. CMS sistem lahko gostuje tako na strežnikih Apache, kot tudi na strežnikih IIS Microsoft (z integriranim PHP jezikom in MySQL podatkovno bazo).

Sistem podpira večuporabniški spletni vmesnik, ki uporabnike deli na tri različne skupine, to so:

- administrator,
- urednik,
- pisec.

Z uporabo sistema CMS dodajajo nove strani ali podstrani ter jih polnijo z vsebino (besedilo, slike, galerije), ki se zapisujejo v podatkovno bazo na strežniku.

Večina enostavnih spletnih strani uporablja na spletni strani elemente, kot so besedila, slike, galerije in kontaktni obrazec. Za večino spletnih strani na spletu, ki uporabljajo odprtokodne sisteme, kot je Joomla je administracija oziroma urejanje vsebine za uporabnike težavno, saj ti sistemi omogočajo veliko funkcij, ki jih običajni uporabniki ne bodo uporabljali in posledično povzročijo zmedo in nepoznavanje okolja sistema.

Sistem je zasnovan na enostavni uporabi za končnega uporabnika, saj je urejanje v vmesniku logično predstavljeno s hierarhijo strani in podstrani na spletni strani tako, da končni uporabnik vedno ve na kateri poziciji se nahaja in na katerem nivoju dodaja ali ureja stran oziroma podstran.

CMS sistem omogoča različne funkcionalnosti z uporabo šestih zavihkov v glavnem meniju:

- E – novice
- Raziskovalec
- Formularji
- Uporabniki
- Vsebina
- mCMS

2.4.1 E – novice

Lastniku spletne strani ali uporabniku spletnega vmesnika omogoča pošiljanje e-novic, ki jih lahko ureja v WYSIWYG urejevalniku besedila. E-poštne naslove prejemnikov e-novic lahko

pridobivamo z obrazcem na spletni strani, ki ga je potrebno integrirati v statično spletno stran tako, da e-poštne naslove zapisuje v podatkovno bazo na strežniku ali pa e-poštni naslov vpišemo ročno v sistemu CMS, kjer so možni tudi popravki ali brisanje e-poštnih naslovov.

V sistemu imamo tudi arhiv e-novic, ki smo jih že v preteklosti pisali, kar nam omogoča uporabo razpošiljanja podobnih novic, ki smo jih že uporabili.

2.4.2 Raziskovalec

Zavihek raziskovalec, ki dostopa do datotečnega sistema na strežniku, kjer gostuje spletna stran in sicer do mape *vsebina*. V tej mapi se običajno nahajajo slike, ki jih lahko naložimo preko modula *galerija* ali *datoteka*. V raziskovalcu imamo dve možnosti pogleda na datoteke v mapi in sicer kot seznam ali kot galerija. V raziskovalcu vidimo ime datoteke, tip datoteke, velikost in podatek ali je datoteka v uporabi na spletni strani, kar nam olajša delo pri iskanju tistih datotek, ki jih ne potrebujemo na našem strežniku in nam zavzemajo prostor. Raziskovalec nam omogoča tudi ogled slikovne datoteke v celotni velikosti, shranjevanje sprememb, brisanje datotek in dodajanje novih datotek na strežnik. Ker so v mapi *vsebina* po večini slikovne datoteke nam raziskovalec omogoča tudi obrezovanje slik na enostaven način principa povleči in spusti (ang. Drag and drop).

2.4.3 Formular

V tem zavihku ima uporabnik možnost nastavitve za kontaktni obrazec oziroma za obrazec o povpraševanju na spletni strani. Uporabnik lahko spreminja polja, ki jih obiskovalci spletne strani preko obrazca pošljejo na e-poštni naslov, ki ga naročnik poda. Na sliki 1 je prikazan vzorec obrazca Oddajte povpraševanje.

Oddajte povpraševanje

Na podlagi izpolnjenega spodnjega obrazca, vam bomo na izbrani kontakt poslali (oz sporočili) ponudbo.

Ponudba za:

težko gradbeno mehanizacijo

nizko gradnjo

avtoprevoznitvo

proizvodnjo, odkup in prodajo humusa

Drugo:

Vaš kontakt - telefon ali e-pošta* - obvezno:

Ime & priimek:

Oddajte povpraševanje

Slika 1: Primer formularja.

2.4.4 Zavihek uporabniki

Sistem nam omogoča tudi upravljanje z uporabniki, ki dostopajo do spletnega vmesnika. V tej menijski izbiri imamo možnost dodajanja novih uporabnikov ter urejanje in brisanje obstoječih uporabnikov, ki dostopajo do spletnega vmesnika CMS. Pri dodajanju novih uporabnikov imamo na voljo tri tipe:

- Pisec – uporabnik, ki ima pravice dodajanja in urejanja vsebine, ki mu ni skrita s strani urednika ali administratorja. Pisec nima možnosti dodajanja novih uporabnikov in brisanja starih uporabnikov.
- Urednik – uporabnik, ki ima pravice dodajanja in urejanja vsebine. Prav tako ima urednik možnost skrivanja strani ali podstrani pred uporabnikom Pisec tako, da jih ta uporabnik ne more spreminjati. Uporabnik tipa Urednik lahko dodaja nove uporabnike tipa Pisec in Urednik.

- Administrator – uporabnik z vsemi pravicami, ki mu omogočajo vse funkcionalnosti kot uporabnikoma Pisec in Urednik, poleg tega pa lahko ta uporabnik dodaja uporabnike vseh tipov (Urednik, Pisec in Administrator). Poleg tega ima uporabnik Administrator še dodaten zavihek (menijsko možnost) ob prijavi v sistem – mCMS, ki je namenjena za kreiranje različnih tipov (pod)strani, katerim lahko dodajamo različne attribute. Uporabnik tipa Administrator je mišljen kot izdelovalec spletne strani, ki bo tudi integral CMS sistem v statično stran HTML.

2.4.5 Zavihek nastavitve

V tej menijski možnosti ima uporabnik, ki je prijavljen s spletni sistem CMS, možnost spremembe svojega gesla oziroma nastavitve novega gesla za svoje uporabniško ime.

2.4.6 Zavihek mCMS

Zavihek mCMS – ta menijska možnost je ob prijavi vidna le uporabnikom z administratorskimi pravicami. Administratorju omogočajo kreiranje tipov strani z različnimi atributi, ki jih ima uporabnik kasneje možnost izbrati ob kreiranju strani ali odstrani.

Atributi:

- tekstovni podatek – za besedila do 255 znakov,
- datum – zapis datuma z uro, predvsem uporabno za novice,
- vsebinski podatek (WYSIWYG) – Kratica opisuje programe, pri katerih uporabniški vmesnik omogoča, da uporabnik med delom vidi, kakšen bo končni dokument oziroma izgled spletne strani, ki jo ureja,
- vsebinski podatek (običajen textbox) – za besedila, ki jih uporabnik ne bo stilsko oblikoval,
- slika – za nalaganje posameznih slik,
- datoteka – za nalaganje datotek,
- galerija – za nalaganje skupin slik, ki so povezane v galeriji – diaproyekcija (ang. slideshow),
- anketa,
- FAQ – pogosto zastavljena vprašanja. Uporabniku doda polje za vpis vprašanja in polje za vpis odgovora,
- formular – za dodajanje polj v kontaktnem obrazcu oziroma v obrazcu za povpraševanje,
- številski podatek – doda uporabniku polje za vnos številskih podatkov. Uporabno za vpis cen, popustov ali drugih številskih polj pri spletnih trgovinah.

Administrator ima možnost določitve imena tipa podstrani, ter dodajanja atributov, sistem omogoča administratorjem tudi kasnejše spreminjanje tipov, dodajanje ali odstranjevanje posameznih atributov in brisanje tipov strani oziroma podstrani. Brisanje tipov strani ni priporočljivo, ker lahko pride do konfliktov, če je tip strani že integriran. Pri dodajanju atributov mora administrator vpisati naziv atributa, ki uporabniku pove, kaj lahko vpisuje v to polje. Prav tako mora administrator pri izbiri atributa izbrati ime spremenljivke preko katere bo možen dostop v PHP kodi do podatkov, ki so zapisani v podatkovni bazi.

2.4.7 Zavihek formularji

V tem zavihku imajo uporabniki možnost izvoza podatkov, ki so jih pridobili iz formularja v XLS datoteko. Datoteko je mogoče pregledovati in obdelovati s programi MS Excel (Microsoft Excel) ali OO Calc (OpenOffice Calc).

3 TEHNOLOGIJE IN ORODJA

Poglavje je namenjeno predstavitvi tehnologij in orodij, ki smo jih izbrali in uporabili pri integraciji spletnega vmesnika v statično HTML stran, ter oblikovanju predloge spletne strani.

3.1 Predstavitev tehnologij

3.1.1 Jezik HTML

HTML (ang. Hyper Text Markup Language) je označevalni jezik za izdelavo spletnih strani in aplikacij in predstavlja osnovo vsakega spletnega dokumenta. S pomočjo HTML-ja ustvarimo strukturo in semantično ureditev dokumenta. HTML kodo lahko pišemo v kateremkoli urejevalniku besedil (beležnica, WordPad, ipd). Jezik je dokaj preprost in se lahko kombinira z ostalimi spletnimi jeziki, kot so PHP, XML, JavaScript. Gradniki jezika HTML so elementi (ang. tags). Samostojni elementi ne potrebujejo zaključka (npr. nov odstavek, nova vrsta), ker vsaka naslednja uporaba tega istega elementa prekliče prejšnjo. Pri ostalih uporabah elementov pa moramo navesti od kje do kje seže učinek tega elementa [8].

3.1.2 Jezik PHP

PHP (ang. Personal Home Page Tool) je odprtokodni skriptni programski jezik, ki se uporablja za strežniško programiranje oziroma za razvoj dinamičnih spletnih vsebin, ki črpajo podatke iz podatkovne baze. Običajno deluje na spletnem strežniku, kjer kot izhod generira HTML kodo s pomočjo PHP jezika [12]. PHP jezik je zelo enostaven za novince v programiranju, kot tudi za programerje, ki so vajeni drugih programskih jezikov, ker omogoča hitro privajanje in enostavno uporabo. Izdelavo spletne strani kreiramo s HTML-jem in PHP-jem. V statično HTML stran vključimo PHP kodo, ki je omejena s posebnimi PHP znaki. Ko obiskovalec odpre stran se začne na strežniku izvajati PHP koda, ki obiskovalcu vrne izpis v HTML obliki, tako da dejansko obiskovalec ne vidi PHP kode v ozadju, za razliko kot pri JavaScript-u.

PHP omogoča odlično povezljivost z različnimi podatkovnimi zbirkami kot so: MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, in Generic ODBC. Najbolj popularna kombinacija je PHP-MySQL. Prav tako kot PHP je tudi MySQL odprtokoden sistem, oba sta na voljo skoraj na vsakem UNIX gostovanju. PHP vsebuje tudi vse funkcije za izvajanje operacij na spletu – povezava na oddaljene strežnike, preverjanje e-pošte preko POP3 ali IMAP protokola, nastavljanje piškotkov (ang. Cookies), preusmerjanje, itd.

3.1.3 Jezik JavaScript

JavaScript je objektni skriptni programski jezik, ki ga je razvil Netscape z namenom, da pomaga spletnim programerjem pri ustvarjanju interaktivnih spletnih strani. Jezik je bil razvit

neodvisno od Jave, vendar si z njo deli številne lastnosti in strukture. JavaScript lahko sodeluje s HTML-kodo in s tem poživi stran z dinamičnim izvajanjem. Podprt je s strani velikih programskih podjetij in kot odprtokodni jezik ga lahko uporablja vsakdo, ne da bi pri tem potreboval licenco. Podpirajo ga vsi novejši spletni brskalniki [9].

Sintaksa jezika JavaScript ohlapno sledi programskemu jeziku C. Prav tako kot C, JavaScript nima vgrajenih vhodno izhodnih funkcij, zato je izvedba teh funkcij odvisna od ponudnika gostovanja oziroma od strežnika na katerem deluje.

JavaScript se veliko uporablja za ustvarjanje dinamičnih spletnih strani. Program se vgradi ali pa vključi v HTML z namenom, da opravlja naloge, ki niso mogoče s samo statično strukturo. Na primer odpiranje novih oken, preverjanje pravilnosti vnešenih podatkov, enostavni izračuni, itd. Na žalost različni spletni brskalniki izpostavijo različne objekte za uporabo. Za podporo vseh brskalnikov je zato potrebno napisati več različic funkcij.

3.1.4 Kaskadne stilske predloge CSS

Stilske predloge CSS (ang. Cascading Style Sheets) so predloge, ki določajo izgled spletnih strani. Z njimi določamo pisavo, velikost črk ter vizualno predstavitev spletne strani. HTML naj bi predstavljal semantično strukturo in smiselno hierarhijo dokumenta, CSS pa ima predstavitveno vlogo. Podpirajo jih vsi novejši spletni brskalniki (Internet Explorer, Netscape Navigator 4.72, Mozilla, Opera, Google Chrom). Spletni brskalniki, ki standarda CSS ne podpirajo, bodo prikazali običajen HTML dokument brez oblikovanja.

CSS je bil izdelan z namenom, da loči vsebino dokumenta (napisanega v HTML-ju ali katerem izmed drugih označevalnih jezikov) od izgleda dokumenta (za katero skrbi CSS). Z uporabo CSS-a lažje spreminjamo izgled strani, stran je bolj fleksibilna in prilagodljiva. Zmanjša se tudi zahtevnost strani, odpravijo se ponavljanja (npr. neprestano oblikovanje naslovov, saj lahko izgled vseh naslovov določimo v predlogi CSS). Za spremembo izgleda strani ni potrebno spreminjati HTML-ja, ampak samo dopolnimo ali popravimo CSS dokument. S CSS-jem si pomagamo tudi pri predstavitvi iste strani v različnih oblikah (za branje, tiskanje, itd). Ko uporabnik prebira vsebino strani je stran prilagojena uporabniku, da jo ta lažje razume. Uporabnik pri branju strani uporablja določene povezave (gumbe), ki ga usmerjajo po dokumentu na strani. Določeni deli teksta in gumbi so posebej označeni, da jih uporabnik lažje najde in uporablja. Posebej označenih gumbov in teksta pa ne potrebujemo, ko želimo stran natisniti. Za odstranitev vseh barvnih gumbov, ki na papirju nebi bili v nikakršno pomoč, skrbi ravno CSS, ki računalniku narekuje, kako brskalniki prikaže stran uporabniku in kako tiskalniku [1].

Problemi, ki so jih povzročali brskalniki s svojstveno implementacijo HTML-ja, se sedaj čutijo tudi pri njihovem razumevanju CSS standarda. Tako imajo razvijalci spletnih strani dvojno delo, da njihova stran izgleda enako v dveh različnih brskalnikih.

3.1.5 MySQL

MySQL (ang. My Structured Query Language) je podatkovna baza, ki nam omogoča shranjevanje raznih podatkov na podatkovnem strežniku v poljubni obliki [7]. Osnovni namen MySQL-a je izvedba specifičnih nalog v podatkovni bazi, ter izvajanje poizvedb nad podatki. Preko strežniških podatkov se poveže s spletno aplikacijo [11].

3.2 Predstavitev orodij

3.2.1 Orodje PhpMyAdmin

PhpMyAdmin je odprtokodna, na brskalniku temeljujoča aplikacija za upravljanje z bazami podatkov MySQL. Njegov grafični uporabniški vmesnik (GUI – Graphic User Interface) omogoča, da se izognemo tradicionalnim ukazno vrstičnim metodam za upravljanjem z bazo MySQL in namesto tega uporabimo kar brskalnik [4].

- Z PhpMyAdmin lahko:
- ustvarimo in izbrišemo podatkovne baze,
- ažuriramo tabele,
- izvršujemo SQL ukaze,
- upravljamo s ključi,
- dodeljujemo pravice in
- izvažamo podatke v različnih formatih.

3.2.2 TextMate

TextMate je urejevalnik izvorne kode na operacijskem sistemu Mac OS X. Komercialni program, ki domuje na MacroMates, ima nekaj samoumevnih lastnosti, kot so na primer, robustnost in preprostost za uporabo. Omogoča tudi nekaj naprednejših funkcij: barvanje sintakse, podvojen pogled. Program je uporaben in primeren tako za manj zahtevne, kot tudi za naprednejše pisce kode [10].

3.2.3 XAMPP

XAMPP ima vse funkcije AMPP (Apache MySQL, PHP, Perl) paketa, ki je ena od redkih nekomercialnih AMPP nakopičenih vmesnih oprem, na voljo v Linux-u. S tesno povezavo, omogoča delovanje funkcij na osebnih domačih straneh, vseh funkcij produkcijske strani, čeprav samo za namene razvoja. Zaradi varnostnih vprašanj XAMPP ni mišljen za uporabo kot produkcijski strežnik [13].

XAMPP ima naslednje prednosti:

- je enostaven za namestitev in nastavitve,
- vsebuje številne uporabne pakete, ki poenostavijo stvari,
- je temeljito preizkušen na operacijskih sistemih SUSE, Red Hat, Mandrake in Debian Linux distribucijah, kot tudi na Windows®, Mac OS in Solaris.

V osnovni paket so vključeni sistem, programiranje in strežniška programska oprema, kot so:

- Apache, najbolj znan spletni strežnik,
- MySQL, odlična, brezplačna in odprtokodna podatkovna baza,
- PHP programski jezik,
- Perl programski jezik,
- ProFTPD, FTP server,
- OpenSSL, za Secure Sockets Layer podporo.

3.2.4 Urejevalnik FCKeditor

FCKeditor je urejevalnik besedila, ki se uporablja v spletnih aplikacijah in na spletnih straneh. Zagotavlja večino skupnih uporabniških funkcij kot drugi urejevalniki besedil, kot so MS Word in OO Write. Z uporabo FCKeditor-ja lahko pišemo besedila, ustvarjamo tabele in slike in še veliko več. Uporabnik ne potrebuje namestitve na svojem računalniku. Vse kar potrebujete za delo je združljiv internetni brskalnik. Urejevalnik FCKeditor je najpogosteje uporabljen urejevalnik v CMS sistemih, ker je enostaven za integracijo in uporabo [5].

3.2.5 Spletni brskalnik Mozilla Firefox

Spletni brskalnik Firefox je brezplačen odprtokodni spletni brskalnik, ki deluje na različnih operacijskih sistemih, kot so Windows, Mac in Linux in v številnih jezikovnih različicah, tudi v slovenskem jeziku. Firefox podpira veliko programskih standardov: HTML, XML, XHTML, CSS, JavaScript, XSL, itd. [7].

3.2.6 Firebug

Firebug omogoča ogled, spremljanje in urejanje HTML-ja, CSS-ja in JavaScript-a v realnem času. Drugače povedano, s pomočjo Firebug-a lahko uredimo spletno aplikacijo direktno iz brskalnika. S tem ne spreminjamo fizične datoteke na disku, ampak samo stran. To nam pomaga hitreje poiskati napako ali dokončati izgled naše spletne strani. Vsebuje tudi razhroščevalnik za JavaScript in lovilec napak [6].

Firebug je zelo uporaben dodatek spletnega brskalnika Mozilla Firefox, saj razvijalcem spletnih aplikacij in spletnih strani olajša delo pri sami obliki aplikacij, kot tudi pri programiranju.

4 ZAHTEVE NAROČNIKA

Naročnik želi, da mu izdelamo predstavitveno spletno stran preko katere želi promovirati prodajo mobilnih telefonov Apple iPhone. Spletna stran naj ima možnost urejanja spletne vsebine v enostavnem CMS urejevalniku.

4.1 Izgled spletne strani

Zahteva naročnika spletne strani je, da ima spletna stran v zgornjem levem kotu logotip, z napisom »SLO iPhone«, črke SLO v barvah slovenske zastave ter v črki O slovenski grb. Beseda »iPhone« mora biti izpisana s takšno pisavo, kot je izpisana na embalaži Applovih iPhonov. Glavni meni na spletni strani mora biti v zgornjem delu poleg logotipa v beli barvi in, v rdeči barvi, ko je obiskovalec na izbrani podstrani. Ozadje glave (ang. header), pa v temnih barvah. Oblika spletne strani naj ima izgled, kot prikazuje skica na sliki 2.



Slika 2: Skica izgleda spletne strani.

4.1.1 Vstopna stran

Na vstopni strani naj bo slika iPhonea 4 v ležeči poziciji, ki vsebuje predstavitev diapozitivov (ang. slide -show), pri katerem se izmenjujejo slike, ki jih uporabnik lahko sam dodaja ali

briše. Poleg ležeče slike iPhona naj bo tudi prostor za besedilo z naslovom. Vsi naslovi naj bodo v modrih odtenkih. Pod grafično obliko telefonskega aparata iPhone naj bo vsebina v treh stolpcih, ki jo uporabnik lahko dodaja ali spreminja. V zadnjem delu vstopne strani pa naj se omogoči uporabniku dodajanja vsebine v dveh stolpcih. Na začetni strani mora biti med nogo in vsebino, nabor najbolj uporabnih aplikacij na iPhonu, le te morajo biti predstavljene s slikami ikon, tako da jih uporabnik lahko dodaja ali odstranjuje.

Noga (ang. footer) na začetni strani naj bo širša, kot na podstraneh, ker bo tukaj vpisanih več podatkov. Na začetni strani naj noga vsebuje eno izmed izjav strank, ki jih lahko vpisujemo v sistemu CMS, ter podatke o podjetju. V nogi se omogočijo tudi povezave do ostalih spletnih strani, tako kot v glavnem meniju, le da je pisava manjša in v rdeči barvi. Barva noge naj bo v temnejših barvah. Noga na ostalih podstraneh naj vsebuje le sekundarni meni v istih barvah kot na začetni strani.

4.1.2 Podstran s tremi stolpci

V sistemu CMS se omogoči dodajanja strani ali podstrani, katerih besedilo je možno razdeliti v tri neodvisne stolpce eden od drugega. Na tej podstrani naj bo tudi možnost dodajanja slik v besedilo. Glavni naslovi naj bodo v modri barvi, kot na vstopni strani.

4.1.3 Podstran z dvema stolpcema

Uporabnik naj ima možnost dodajanja podstrani ali strani, katerih besedilo je možno razdeliti v dva neodvisna stolpca. Prav tako kot na strani s tremi stolpci naj ima uporabnik možnost dodajanja slik med besedilo. Glavni naslovi naj bodo v modri barvi, kot na vstopni strani.

4.1.4 Podstran z galerijo

Vsebuje možnost dodajanja strani ali podstrani z modulom galerija. Poleg galerije naj ima uporabnik tudi možnost dodajanja besedila, ki ga lahko sam stilsko ureja. Besedilo, ki se prikazuje na tej podstrani naj bo prikazano nad galerijo.

4.1.5 Podstran kontakti

Na tej podstrani naj bodo vsi kontaktni podatki podjetja:

- naziv podjetja,
- naslov podjetja,
- kraj in poštna številka,
- stacionarna telefonska številka podjetja,
- mobilna telefonska številka podjetja,

- elektronski poštni naslov podjetja.

Poleg zgoraj naštetih podatkov naj ta stran vsebuje tudi zemljevid, kje se podjetje nahaja, ter kontaktni obrazec, preko katerega lahko obiskovalci pošljejo povpraševanje ali splošna vprašanja na elektronski poštni naslov podjetja. Kontaktni obrazec naj vsebuje polja za vpis imena, e-poštnega naslova pošiljatelja ter polje za vnos sporočila. Vsa polja naj bodo obvezna, kar pomeni, da naj pred pošiljanjem sporočila pošiljatelja opozori, da mora izpolniti obvezna polja. Polje za vnos elektronskega poštnega naslova naj se ob pošiljanju tudi preveri, če je elektronski poštni naslov zapisan v pravilni obliki. Ob uspešnem poslanem sporočilu naj bo obiskovalec obveščen v obliki elektronskega sporočila.

4.2 Funkcionalne zahteve

Temeljne funkcionalne zahteve izdelave spletne strani z urejevalnikom spletne vsebine:

- kreiranje novih strani in podstrani,
- urejanje besedil podobnim urejanju v urejevalnikih besedil na vseh podstraneh,
- možnost dodajanja novih uporabnikov,
- hranjenje vseh kreiranih strani v podatkovni bazi.

5 INTEGRACIJA CMS SISTEMA V STATIČNO SPLETNO STRAN

Za integracijo CMS sistema v statično spletno stran, bomo najprej poiskali predlogo spletne strani, ki bo vsebovala največ elementov, ki jih potrebujemo za izdelavo spletne strani po zahtevah namišljenega podjetja za prodajo iPhoneov.

Predlogo spletne strani, ki je v dodatku A in smo jo pridobili na spletni strani Template World smo morali spremeniti tako, da čim bolj ustreza danim zahtevam za izdelavo spletne strani. Večino sprememb smo morali opraviti v CSS datotekah, ki določajo izgled spletne strani in nekaj sprememb v HTML datotekah. Najprej se bomo osredotočili na vizualni izgled spletne strani v statičnem HTML označevalnem jeziku, kar pomeni, da je pred dejansko integracijo potrebno predelati predlogo spletne strani, ki smo si jo izbrali tako, da bo ustrezala zahtevam naročnika. Na sliki 3 sta prikazani izbrana predloga in končna oblika statične spletne strani, ki jo bomo uporabili za integracijo CSM sistema.



Slika 3: Primerjava strani iz predloge in preoblikovane strani po zahtevah naročnika.

Ko imamo izdelano statično stran, se lotimo kreiranja podatkovne baze, ki temelji na MySQL podatkovni bazi. Bazo bomo kreirali z SQL stavkom, tako da bomo dejansko uvozili tabele, ki jih potrebujemo za integracijo CSM sistema.

Za predstavitev integracije CSM sistema postavimo spletno stran na lokalnem virtualnem Apache strežniku, tako da moramo prenesti datoteke, ki jih potrebujemo za delovanje CMS sistema na lokalne direktorije. Pred dejanskim programiranjem moramo dodati določenim datotekam še dostopne podatke do podatkovne baze, po spremembi teh datotek pa moramo preveriti še vse povezave s podatkovno bazo oziroma testirati ali smo dostopne podatke pravilno vpisali v datoteke in če lahko dostopamo do CMS sistema preko strežnika.

Ko imamo povezavo med podatkovno bazo in statično spletno stranjo ter imamo dostop do CMS sistema, se lahko lotimo nadaljnje integracije. Najprej moramo biti v sistemu CMS prijavljeni kot administrator, v zavihku mCMS kreirati nove tipe (pod)strani, ki jih bomo potrebovali za nadaljnjo integracijo. Postopek kreiranja tipov lahko v zavihku mCMS opravljamo sproti po potrebi, ko bomo vedeli katere atribute bomo dejansko potrebovali na straneh oziroma podstraneh. S kreiranimi tipi strani oziroma podstrani se bomo lahko lotili kreiranja datoteke *index.php*, ki bo vstopna stran naše spletne strani. V tej datoteki bomo opravili večino programiranja v jeziku PHP in klicev podatkov iz podatkovne baze.

5.1 Izbira statične HTML predloge

Za primer integracije spletnega vmesnika smo si izbrali že obstoječo HTML stran z elementi CSS-a, JavaScripta in JQuery-ja. Spletno stran smo pridobili na spletni strani Template World. Spletne strani oziroma predloge spletnih strani na tej strani so plačljive in zaščitene z licenco, ki prepoveduje:

- nadaljnjo prodajo in izdajo,
- prisvojitve oblikovanja oziroma izjave da je bila oblika spletne strani izdelana z vaše strani ali strani vaših sodelavcev.

Licenca TemplateWorld.com dovoljuje:

- prosto uporabo predloge za vašo osebno spletno stran,
- prosto uporabo predloge za vaše podjetje / spletno stran podjetja in
- prosto uporabo predloge za spletne strani vaših strank.

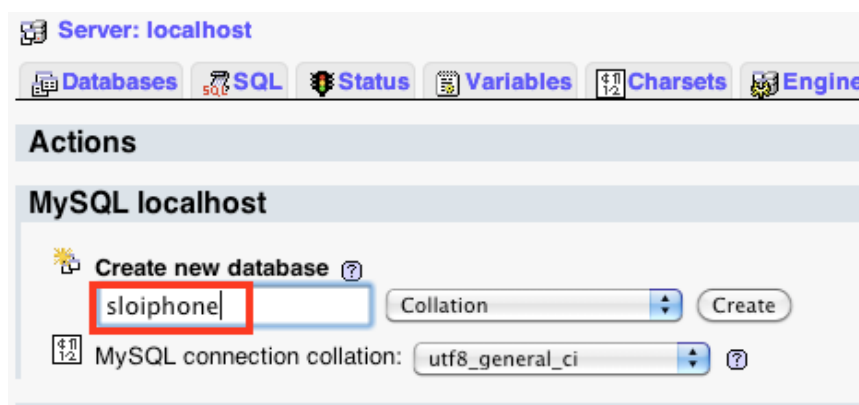
Pri izbiri predloge smo se osredotočili na nekaj podstrani, ki jih bomo potrebovali za izdelavo spletne strani.

5.2 Kreiranje podatkovne baze

Podatkovna baza bo delovala na Apache-jevem strežniku. Upravljali jo bomo z orodjem phpMyAdmin.

5.2.1 Kreiranje baze

Ker za integracijo CMS sistema uporabljamo lokalni virtualni strežnik se za kreiranje podatkovne baze prijavimo v phpMyAdmin. Podatkovno bazo kreiramo tako kot prikazuje slika 4, za ime baze izberemo *sloiphone*, sledi še dodajanje uporabnika za to bazo.



Slika 4: Kreiranje nove podatkovne baze v phpMyAdmin.

5.2.2 Kreiranje uporabnika podatkovne baze

Uporabnika, ki dostopa do podatkovne baze smo dodali v phpMyAdminu. Za kreiranje uporabnika baze imamo dve možnosti in sicer preko SQL stavka ali pa preko phpMyAdmin vmesnika za dodajanje uporabnikov v bazo. Na sliki 5 je prikazano dodajanje uporabnika preko vmesnika na sliki 6 pa dodajanje preko SQL stavka v phpMyAdminu.

Slika 5: Kreiranje uporabnika podatkovne baze preko vmesnika v phpMyAdminu.

Podatke o uporabniku bomo kasneje uporabili v programskem jeziku PHP, s katerim se bomo povezovali na to podatkovno bazo, zato si je potrebno te podatke zapomniti (ime strežnika, uporabnika podatkovne baze in geslo). Ker uporabljamo lokalni virtualni strežnik Apache bo ime strežnika – *localhost*.

```
GRANT ALL
ON sloiphone.*
TO JURE
IDENTIFIED BY 'diploma99'
```

Slika 6: Dodajanje pravic uporabniku z SQL ukazom.

5.2.3 Kreiranje tabel v podatkovni bazi

V podatkovni bazi za integracijo CMS sistema bomo potrebovali 26 tabel, ki jih bomo uvozili z SQL stavkom, ki je priložen v dodatku D. Opisi tabel v podatkovni bazi, ki jih bomo največkrat uporabljali so opisani v nadaljevanju.

5.2.3.1 Stran

V tabeli *Stran* imamo v podatkovni bazi shranjene vse podatke, ki jih potrebujemo za integracijo sistema v statični HTML, ki se sklicujejo na stran oziroma podstran na spletni strani, ki jo pridobivamo iz podatkovne baze.

Polje	Tip	Null	Default	Extra
<i>Stranid</i>	Int(11)	No	None	auto_increment
<i>Parent</i>	Int(11)	No	0	
<i>Uporabnikid</i>	Int(11)	No	0	
<i>Tipid</i>	Int(11)	No	0	
<i>Naslov</i>	Varchar(255)	Yes	NULL	
<i>Modified</i>	Int(11)	No	0	
<i>Red</i>	Int(11)	No	0	
<i>Deleted</i>	Int(11)	No	0	
<i>Skrij</i>	Int(11)	No	0	
<i>Zaklenjen</i>	Int(11)	No	0	
<i>Sellable</i>	Int(11)	No	0	

Opis polj v tabeli *Stran*:

Stranid – polje, ki nam pove unikatni ključ strani, tipa integer in se povečuje samodejno oziroma avtomatsko.

Parentid – polje, ki nam poda starša strani, tipa integer. Uporabno, ko potrebujemo nabor vseh otrok določene strani iz podatkovne baze.

Primer SQL stavka: Izpiši vse strani za glavno navigacijo spletne strani, ki niso skrite.

```
SELECT * FROM stran WHERE parent = 2 AND skrij = 0
```

Na sliki 7 je prikazan izpis rezultatov zgornje poizvedbe.

		stranid	parent	uporabnikid	tipid	naslov	modified	red	deleted	skrij	zaklenjen	sellable
<input type="checkbox"/>			3	2	1	2	Gosti	1304430201	1	0	0	0
<input type="checkbox"/>			4	2	1	2	Novice	1304409437	2	0	0	0
<input type="checkbox"/>			15	2	1	2	Galerija	1304932386	3	0	0	0
<input type="checkbox"/>			16	2	1	4	O nas	1304928472	4	0	0	0

Slika 7: Izpis SQL rezultatov poizvedbe iz podatkovne baze.

Uporabnikid – polje, ki nam poda id uporabnika, ki je nazadnje dostopal do tabele stran.
Uporabnikid je ključ tabele uporabnik

Tipid – polje, ki nam poda id tipa strani. Tip strani nastavljam v CMS sistemu v zavihku mCMS.

Naslov – polje, ki nam vrne ime oziroma naslov strani.

Red – polje, ki nam vrne vrstni red – zaporedno številko dodane spletne strani.

Primer SQL stavka: Izpiši vse id strani in naslove strani s staršem 2, ki niso skrite in izbrisani ter jih izpiši po vrstnem redu.

```
SELECT stranid, naslov FROM stran WHERE parent = 2 AND deleted = 0 AND
skrij = 0 ORDER BY red ASC
```

Na sliki 8 so prikazani rezultati poizvedbe strani za glavno navigacijo.

			stranid	naslov
<input type="checkbox"/>			20	iPhone 4
<input type="checkbox"/>			14	Nakup
<input type="checkbox"/>			21	Galerija
<input type="checkbox"/>			17	Kontakt

Slika 8: Izpis rezultatov poizvedbe za glavni meni.

Deleted – nam vrne 1, če je bila stran s strani CMS vmesnika izbrisana, ter 0 če ni bila izbrisana.

Skrij – nam vrne 1, če je stran skrita (opcija v CMS vmesniku), ter 0 če stran ni skrita.

Zaklenjen – nam vrne 1, če je stran zaklenjena pred urejanjem uporabnika tipa pisec, ter 0 če stran ni zaklenjena.

Sellable – nam vrne 1, če je stran artikel v spletni trgovini, ter 0 če ni artikel.

5.2.3.2 Tip

V tabeli *Tip* imamo vse tipe strani in podstrani, ki jih administratorji lahko kreirajo v zavihku mCMS v spletnem vmesniku.

Polje	Tip	Null	Default
<i>Tipid</i>	Int(11)	No	None
<i>Naziv</i>	Varchar(255)	Yes	NULL
<i>Deleted</i>	Int(11)	no	0

Opis polj v tabeli *Tip*:

Tipid – poda unikatni ključ tipa strani, tipa integer in se povečuje samodejno oziroma avtomatsko.

Naziv – poda naziv določenega tipa, ki smo ga dodelili v spletnem vmesniku v zavihku *mCMS*.

Deleted – poda podatek ali je bil ta tip strani izbrisan ali ne. (0 tip ni bil izbrisan, 1 tip je bil izbrisan).

5.2.3.3 Galerija

V tabeli *galerija* imamo vse galerije, ki se pojavljajo na straneh.

Polje	Tip	Null	Default
<i>Galerijaid</i>	Int(11)	No	None
<i>Atributid</i>	Int(11)	No	0
<i>Stranid</i>	Int(11)	No	0
<i>Red</i>	Int(11)	No	0
<i>File</i>	Varchar(255)	Yes	NULL
<i>Opis</i>	Varchar(255)	Yes	NULL

Opis polj v tabeli *Galerija*:

Galerijaid – nam poda unikatni ključ galerije, tipa integer in se povečuje samodejno.

Atributid – nam poda id atributa (sekundarni ključ).

Stranid – nam poda id strani na katero je galerija vezana.

Red – nam poda vrstni red galerije.

File – nam pove pot datoteke (slike) v galeriji.

Opis – nam poda opis vsake datoteke (slike) v galeriji.

5.2.3.4 Atribut

V tabeli *Atribut* imamo vse attribute, ki smo jih uporabili pri kreiranju tipov strani v sistemu CMS.

Polje	Tip	Null	Default
<i>Atributid</i>	Int(11)	No	None
<i>Tipid</i>	Int(11)	No	0
<i>Opis</i>	Varchar(255)	Yes	NULL
<i>Vrsta</i>	Int(11)	No	0
<i>Red</i>	Int(11)	No	0
<i>Deleted</i>	Int(11)	No	0
<i>Spremenljivka</i>	Varchar(255)	No	

Opis polj v tabeli *Atribut*:

Atributid – nam poda unikatni ključ atributa, tipa integer in se povečuje samodejno.

Tipid – poda id tipa pri katerih je ta atribut uporabljen.

Opis – nam poda opis atributa, ki ga administrator vpiše v zavihku mCMS spletnega vmesnika.

Vrsta – nam vrne številko (int), ki pove vrsto atributa. Uporabljamo:

- 1 – datum.
- 2 – vsebinski podatek z WYSIWYG urejevalnikom.
- 3 – vsebinski podatek z običajnim tekstovnim poljem.
- 4 – slika.
- 5 – datoteka.
- 6 – galerija.
- 7 – anketa.
- 8 – FAQ – pogosto zastavljena vprašanja.
- 9 – formular.

Red – nam vrne številko vrstnega reda atributa.

Deleted – nam poda podatek ali je bil ta tip strani izbrisan ali ne. (0 tip ni bil izbrisan, 1 tip je bil izbrisan).

Spremenljivka – nam pove ime spremenljivke, ki jo je administrator dodal atributu v spletnem vmesniku.

5.3 Prenos datotek na strežnik

Za integracijo CMS sistema za statično spletno stran moramo najprej prenesti vse datoteke, ki jih potrebujemo za statično spletno stran, ki smo jo predelali oziroma dodelali iz predloge. Poleg tega moramo dodati datoteke CMS sistema v mapo na lokalnem virtualnem strežniku, kjer imamo že datoteke naše statične spletne strani.

Ker uporabljamo lokalni virtualni strežnik XAMPP na Mac OS X operacijskem sistemu se ta mapa nahaja v `/Applications/XAMPP/htdocs/sloiphone`. Po prenosu datotek, ki jih potrebujemo za statično spletno stran brez integriranega CMS vmesnika, moramo prenesti v to mapo še datoteke in mape, ki jih potrebujemo za samo delovanje spletnega vmesnik. Te datoteke in mape so:

- mapa *admin*,
- datoteka *escape.php*,
- datoteka *database_connect.php*,
- datoteka *funkcije.php*.

V mapi *admin* se nahajajo še ostale podmape in datoteke, ki so prikazane na sliki 9.

actions	img	style-ie7.css
akcije	include.php	style-popup.css
anketa.php	index.php	style.css
crop	jscript	t-narocila.php
enovice.php	login.css	t-nast.php
error_log	mcms.php	t-upor.php
export_xls.php	narocila.php	test.php
faq.php	nastavitve.php	uporabniki.htm
fckeditor	php.ini	uporabniki.php
footer.php	popup	vsebina.php
formular.php	raziskovalec.htm	zavihki.php
formularji.php	raziskovalec.php	
galerija.php	raziskovalec2.htm	

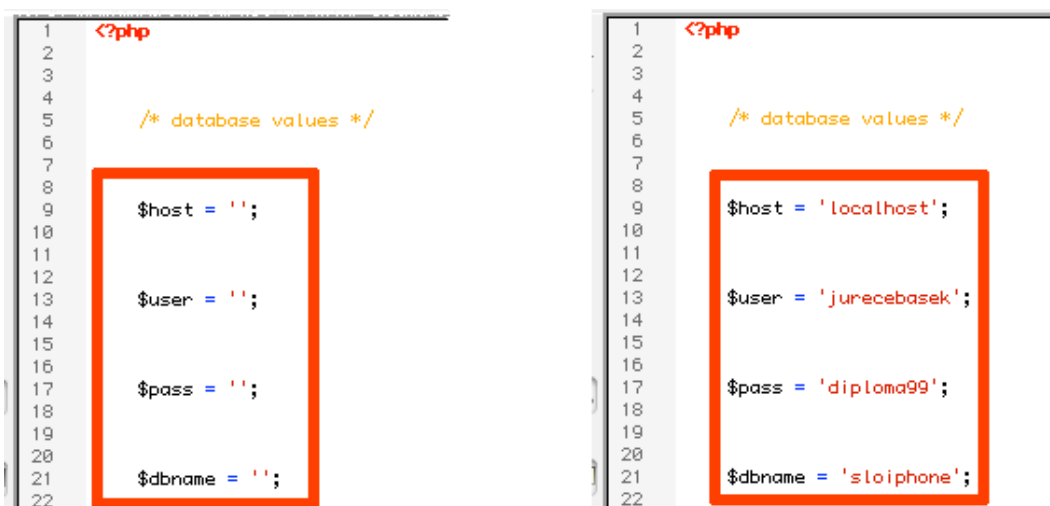
Slika 9: Datoteke, ki so potrebne za integracijo vmesnika.

5.4 Spreminjanje datotek za dostop do podatkovne baze

Da preverimo ali nam povezava do podatkovne baze v MySQL-u deluje, moramo v dve datoteki tipa php, vpisati podatke uporabnika, ki smo mu dodelili pravice za dostop pri kreiranju podatkovne baze.

Datoteka *database_connect.php* služi povezavi spletne strani na podatkovno bazo iz katere poizvedujemo podatke, ki smo jih preko CMS sistema vpisali v to podatkovno bazo. Spremembe v tej datoteki se začnejo v deveti vrstici kode, tako da vpišemo vse podatke uporabnika, da se lahko povežemo na podatkovno bazo.

V spremenljivko *\$host* bomo v našem primeru, ker delamo na virtualnem lokalnem strežniku vpisali vrednost »localhost«, v spremenljivko *\$user* vrednost »jurecebasek«, v spremenljivko *\$pass* bomo vpisali vrednost gesla za uporabnika *jurecebasek* in sicer »diploma99«, ter v spremenljivko *\$dbname* bomo vpisali ime podatkovne baze na katero se povezujemo »sloiphone«. Primer vpisa podatkov v datoteki *database_connect.php* in datoteki *admin/include.php* je na sliki 10.



Slika 10: Prikaz vpisa podatkov za dostop do podatkovne baze.

Datoteka *admin/include.php* prav tako služi za povezovanje na podatkovno bazo in je vključena v vsaki skripti. Najprej zažene »session«, potem se poveže s podatkovno bazo. Ta povezava služi povezovanju spletnega vmesnika CMS na podatkovno bazo, tako da dejansko lahko deluje CMS sistem neodvisno. Kot v datoteki *database_connect.php* moramo tudi v datoteki *admin/include.php* vpisati v spremenljivke *\$host*, *\$user*, *\$pass* in *\$dbname* podatke uporabnika za povezavo na podatkovno bazo.

5.5 Kreiranje tipov strani v CMS sistemu

V spletnem vmesniku CMS je potrebno pred integracijo v izvorni kodi statične spletne strani najprej kreirati tipe strani, ki jih bodo lahko uporabniki sistema dodajali. Kreiranje tipov

strani in podstrani lahko v spletnem vmesniku kreira le uporabnik sistema tipa administrator v zavihku *mCMS*, kot je prikazano na sliki 11.



Slika 11: Zavihek mCMS v sistemu CMS.

V zavihku imamo v začetni poziciji samo možnost dodajanja novega tipa strani, pri kateri moramo najprej vpisati naziv tipa. Naziv tipa spletne strani je najbolje smiselno poimenovati, tako, da bodo lahko uporabniki, ki bodo uporabljali spletni vmesnik CMS za dodajanje novih strani in podstrani točno vedeli kateri tip bodo dodali in kaj jim ta tip spletne strani po opisu sodeče omogoča. Ko kreiramo nov tip strani, imamo možnosti dodajanja atributov vsakemu tipu strani.

Zaradi boljše orientacije po spletnem vmesniku, ki uporabnikom povzročajo največ težav, bomo kreirali tip strani *navigacija*, ki bo služil le navigacijskim možnostim v zavihku vsebina. Kreirali bomo tri navigacijski strani:

- *Vsebina vstopne strani*, kjer bo nekaj več tipov strani.
- *Vsebina* – tu bodo dejansko nanizane vse podstrani naše spletne strani
- *Izjave strank* – tu bodo nanizane izjave stank

Tip strani *navigacija* bo brez atributov, ker nam bo služil le zaradi boljše orientacije po sistemu.

5.5.1 Kreiranje tipa strani Vstopna stran

Ker se bomo najprej lotili izdelave tipov vstopne strani na spletni strani, bomo to stran razdelili na več segmentov, saj je ta stran najobširnejša in bo obiskovalcem prikazovala največ informacij in podatkov. Če v grobem razdelimo vstopno spletno stran brez noge in glave spletne strani vidimo, da lahko spletno stran razdelimo na sedem segmentov, kot so prikazani na sliki 12.



Slika 12: Razdelitev vstopne strani na segmente.

Pri kreiranju novega tipa strani imamo v spletnem sistemu CMS možnost dodajanja atributov vsakemu tipu strani. Vsaki strani lahko dodamo poljubno število atributov, vendar je zaradi preglednosti najbolje, da omejimo število atributov tako, da se lahko navadni uporabniki znajdejo pri dodajanju strani. Za dodajanje atributa moramo vpisati naziv atributa, ime spremenljivke atributa, ter iz padajočega menija (ang. drop-down menu) izbrati atribut.

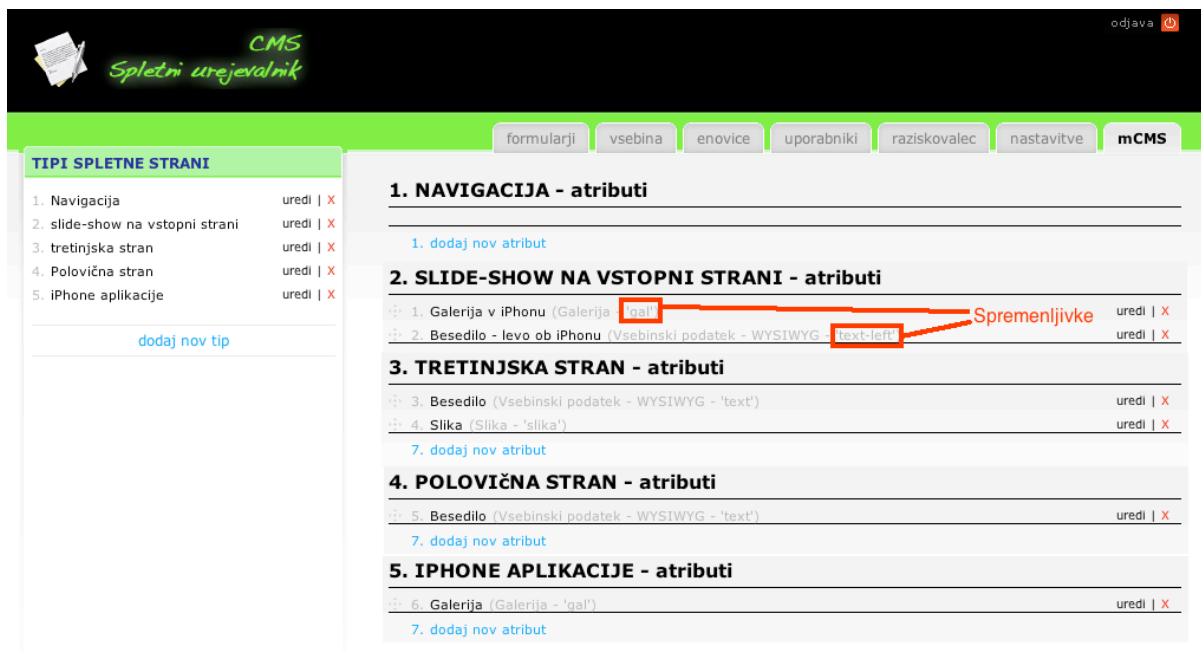
Za vstopno stran bomo kreirali štiri različne tipe strani, saj vidimo da segmenti št. 2, 3 in 4 uporabljajo iste tipe strani, ter segmenta št. 5 in 6 tudi isti tip strani.

Pri segmentu št. 1 vidimo, da bomo potrebovali tri različne attribute in sicer polje za vnos naslova, polje za vnos besedila pod naslovom – WYSIWYG, ter galerijo za predstavitev diapozitivov (ang. slide-show). Pri kreiranju tipov strani se lahko izognemo dodajanju polja za naslove, saj lahko pri dodajanju spletne strani v zavihku *Vsebina* uporabimo polje »naslov« tudi za izpis dejanskih naslovov na spletni strani.

Pri segmentih št. 2, 3 in 4 bomo potrebovali attribute *polje za vnos besedila – WYSIWYG* in *slika*. Če katerega od atributov, ki so na voljo na strani, uporabnik ne uporabi, se ta ne bo izpisal. Tako bi dejansko vsaki podstrani lahko dodali vse attribute, vendar bi urejanje strani postalo nepregledno in moteče za uporabnike, tako da se osredotočimo le na tiste attribute, ki jih bo uporabnik dejansko potreboval in urejal v spletnem vmesniku.

Pri segmentih št. 5 in 6 bomo potrebovali le urejevalnik besedila WYSIWYG, ki nam omogoča tudi dodajanje slike med besedilom in pozicioniranje teh slik.

Za segment št.7 bomo prav tako potrebovali le en atribut – galerijo za dodajanje slik, saj vnos besedila za naslov ne potrebujemo, ker imamo možnost dodajanja naslova strani v zavihku *Vsebina*. Po vnosu vseh atributov za vsebino na vstopni strani izgleda zavihkec *mCMS*, kot prikazuje slika 13.



Slika 13: Zavihkec mCMS po vnosu vseh podstrani in atributov.

5.5.2 Kreiranje tipa strani Podstran

Spletna stran bo zasnovana tako, da bodo lahko uporabniki dodajali podstrani in urejali vsebino vstopne strani. Za integracijo podstrani bomo kreirali tri različne tipe podstrani, ki jih bomo potrebovali, da bodo lahko uporabniki dodajali vsebino.

Podstrani bomo razdelili na:

- Podstran s tremi kolonami – sestavljena bo iz treh kolon besedila, katere je možno urejati preko spletnega vmesnika CMS. Tej strani bomo dodali nov tip s tremi atributi *Vsebinski podatek – WYSIWYG*.
- Podstran s dvema kolonoma – sestavljena bo iz dveh kolon besedila, katere je možno urejati preko spletnega vmesnika CMS. Tej strani bomo dodali nov tip z dvema atributi *Vsebinski podatek – WYSIWYG*.
- Podstran z galerijo – vsebovala bo atribut *Vsebinski podatek – WYSIWYG* za vnos besedila, ter atribut *Galerija* za nalaganje slik v galerijo spletne strani. Galerija bo na spletni strani prikazana kot predstavitev diapozitivov (ang. slide-show).

Na sliki 14 so prikazani vsi atributi zgornjih podstrani, ki jih bomo uporabili na naši spletni strani.

The screenshot shows the CMS interface with a navigation bar at the top containing 'formularji', 'vsebina', 'enovice', 'uporabniki', 'raziskovalec', 'nastavitve', and 'mCMS'. On the left, a sidebar titled 'TIPI SPLETNE STRANI' lists eight page types, with 'Podstran 3 kolone', 'Podstran 2 koloni', and 'Podstran z galerijo' selected. The main content area displays the configuration for these three types:

- 6. PODSTRAN 3 KOLONE - atributi**
 - 7. Besedilo - levo (Vsebinski podatek - WYSIWYG - 'text-left')
 - 8. Besedilo - center (Vsebinski podatek - WYSIWYG - 'text-center')
 - 9. Besedilo - levo (Vsebinski podatek - WYSIWYG - 'text-left')
- 7. PODSTRAN 2 KOLONI - atributi**
 - 10. Besedilo - levo (Vsebinski podatek - WYSIWYG - 'text-left')
 - 11. Besedilo - desno (Vsebinski podatek - WYSIWYG - 'text-right')
- 8. PODSTRAN Z GALERIJO - atributi**
 - 12. Besedilo (Vsebinski podatek - WYSIWYG - 'text')
 - 13. Galerija (Galerija - 'gal')

Each attribute entry includes a 'uredi | X' link and a '14. dodaj nov atribut' link.

Slika 14: Podstrani in atributi v CMS vmesniku.

5.5.3 Kreiranje tipa strani Kontakt

Ta tip strani bo namenjen za stran s kontaktnimi podatki podjetja, ter obrazcem preko katerega lahko uporabniki spletne strani pošljejo povpraševanje na elektronski poštni naslov podjetja. Na sliki 15 so prikazani vsi atributi strani tip *kontakt* s spremenljivkami.

TIPI SPLETNE STRANI	
1. Navigacija	uredi X
2. slide-show na vstopni strani	uredi X
3. tretinjska stran	uredi X
4. Polovična stran	uredi X
5. iPhone aplikacije	uredi X
6. Podstran 3 kolone	uredi X
7. Podstran 2 koloni	uredi X
8. Podstran z galerijo	uredi X
9. Kontakti	uredi X

9. KONTAKTI - atributi	
14. Naziv (Tekstovni podatek - do 255 znakov - 'naziv')	uredi X
15. Ulica (Tekstovni podatek - do 255 znakov - 'ulica')	uredi X
16. Mesto, Poštna številka (Tekstovni podatek - do 255 znakov - 'mesto')	uredi X
17. Država (Tekstovni podatek - do 255 znakov - 'drzava')	uredi X
18. GSM (Tekstovni podatek - do 255 znakov - 'gsm')	uredi X
19. TEL (Tekstovni podatek - do 255 znakov - 'tel')	uredi X
20. FAX (Tekstovni podatek - do 255 znakov - 'fax')	uredi X
21. E-mail (Tekstovni podatek - do 255 znakov - 'mail')	uredi X
22. dodaj nov atribut	

Slika 15: Atributi za podstran tipa kontakt.

5.5.4 Kreiranje tipa strani Izjave naših strank

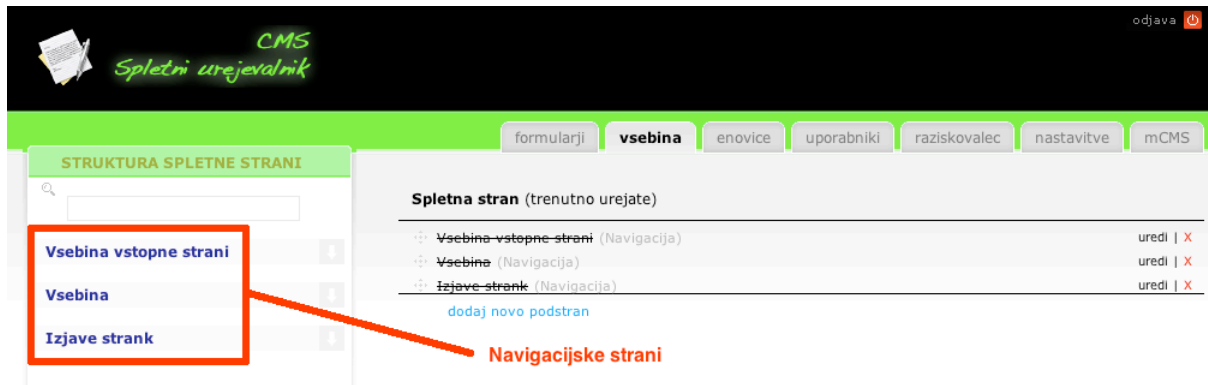
Ker imamo na vstopni strani v nogi spletne strani prostor za izjave naših strank, bomo v sistemu CMS kreirali nov tip strani za vnos teh podatkov. Naročnik želi dinamično spletno stran, zato bomo dodali možnost nalaganja več izjav strank, ki pa se bodo na vstopni strani prikazovale naključno. Za ta tip strani bomo potrebovali le vnosno polje za vnos besedila – *običajen textbox* kamor bodo uporabniki sistema lahko vpisovali izjave svojih strank.

5.6 Kreiranje strukture strani v CMS sistemu

Strukturo spletne strani kreiramo v zavihku *Vsebina*, kjer lahko dodajamo vse tipe strani, ki smo jih kreirali kot administratorji v zavihku *mCMS*. Kasnejšim uporabnikom želimo omogočiti kar najbolj pregleden urejevalnik spletne vsebine, zato bomo uporabili tri tipe strani *navigacija* za boljše orientiranje po vmesniku CMS.

Dodali smo navigacijske podstrani, ki so prikazane na sliki 16. Vstopna stran spletne strani, ki jo izdelujemo je najobširnejša z največ elementi, zato smo ji dodelili navigacijski tip, kateremu bomo dodajali ostale tipe, ki jih potrebujemo za vstopno stran. Drugi navigacijski tip strani bo služil za vsebino. Podstrani navigacijske strani *Vsebina* bodo vse podstrani na spletni strani, ki se bodo prikazovale na spletni strani v zgornjem meniju. Zadnji

navigacijski tip strani je namenjen izjavam strank. Podstrani tega navigacijskega tipa bodo strani tip *Izjave strank*, ki smo ga kreirali v zavihku *mCMS*.



Slika 16: Navigacijske strani v CMS vmesniku.

Ko imamo kreirane navigacijske strani, se lahko lotimo dodajanja strani, ki bodo vidne na spletni strani. Strani dodajamo v vmesniku CMS tako, da se najprej postavimo na navigacijsko stran, kateri želimo dodati podstran in izberemo povezavo *dodaj novo podstran*. V vmesniku CMS se nam prikaže nov del strani, kamor moramo vpisati naslov podstrani, ki jo dodajamo, ter iz spustnega menija (ang. drop-down) izbrati tip strani, ki jo bomo dodali. Ko podstran dodamo, se ta pojavi na levi strani vmesnika CMS, pod navigacijsko stranjo, kateri smo jo dodali. Vse novo kreirane strani so privzeto skrite, kar pomeni, da se stran na spletni strani ne bo prikazala, dokler ji ne odstranimo statusa skrita stran.

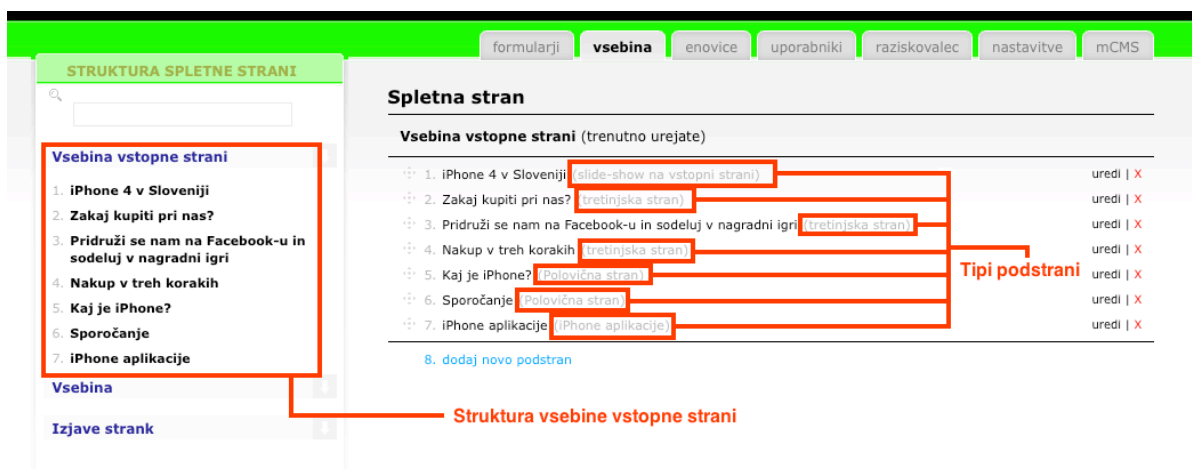
Možnost skrivanja je zelo uporabna v tovrstnih sistemih, ker omogoča, da uporabniki spletno stran uredijo tako, kakor so si jo zamislili pred objavo na spletu. Obstaja možnost, da bi lahko obiskovalec spletne strani ravno v tem času kliknil na novo kreirano stran, ki ne bi vsebovala nobene vsebine, kar bi pri obiskovalcu sprožilo vprašanje ali se mu stran ne prikazuje pravilno ali, da je lastnik oziroma administrator spletne strani pozabil vpisati vsebino. S skrivanjem strani se izognemo takim situacijam.

Primer: navigacijski strani *Vsebina vstopne strani* bomo dodali tiste podstrani, ki jih moramo prikazovati na vstopni strani spletne strani. Na sliki 17 je prikazano dodajanje prve podstrani na vstopni strani tipa *slide-show na vstopni strani*.



Slika 17: Dodajanje podstrani na vstopni strani.

Po dodanih vseh potrebnih podstraneh v navigacijskem tipu strani *Vsebina vstopne strani* v sistemu CMS je struktura vstopne strani prikazana na sliki 18. Na vsako od dodanih podstrani smo vnesli naslov podstrani ter demo besedilo.



Slika 18: Vstopna stran v CMS sistemu z vsemi podstranmi.

Na navigacijski strani *Vsebina* bomo dodali podstrani

- iPhone 4,
- Nakup,
- Galerija in

- Kontakt.

Vsaka od podstrani bo drugačnega tipa (Podstran 3 kolone, Podstran 2 koloni, Podstran z galerijo in tip Kontakti). Ko vnesemo vse podstrani v navigacijskem tipu strani *Vsebina* z naslovi in testnim besedilom izgleda v sistemu za urejanje spletne vsebine CMS tako, kot je prikazano na sliki 19.



Slika 19: Podstrani v navigacijskem tipu vsebina.

Pri zadnji navigacijski strani *Izjave strank* bomo vnesli dve podstrani izjav strank tipa *Izjave strank*. Dve strani smo dodali, ker se bodo na vstopni strani spletne strani prikazovale naključno. Nadaljnje izjave strank lahko uporabniki sistema dodajajo na isti način.

5.7 Izvedba integracije CMS sistema v PHP različici

V zadnji fazi integracije spletnega vmesnika bomo morali v statično HTML spletno stran implementirati povezavo s CMS sistemom, preko katerega uporabnik polni podatkovno bazo katere vsebino moramo prikazati na spletni strani.

Programiranja spletne strani se lahko lotimo na več načinov. Eden od načinov je, da kreiramo za vsak tip strani nov PHP dokument, ter se v tem dokumentu osredotočimo le na določeno podstran oziroma stran in začetni dokument običajno *index.php* povežemo z drugimi dokumenti ob akciji, ko uporabnik klikne na drugo stran. Drug način, ki ga bomo uporabili tudi pri tej implementaciji je, da lahko celotno spletno stran z vsemi stranmi programiramo v eni sami datoteki PHP. Pri tem načinu lahko tudi izkoristimo hitrejšo delovanje na strežniku same spletne strani, saj se dejansko nekateri deli spletne strani ponavljajo in jih ni potrebno nalagati pri vsakem kliku na stran, ampak lahko izpisujemo samo dejansko vsebino, ki se spreminja.

Ker se bomo lotili programiranja spletne strani v eni sami datoteki, bomo to datoteko razdelili v strukturo, kot je prikazano v spodnji psevdo kodi na sliki 20.

```

<?php
    povezava s podatkovno bazo
    dodelitev tipov strani
?>

<?php
    izpis glave spletne strani
    izpis menija
?>

<?php
    če je tip spletne strani tipa vstopna stran
    {
        izpis strani tipa vstopna stran z vsemi klici iz podatkovne baze
        izpis noge vstopne strani
    }
?>

<?php
    če je tip spletne strani tipa podstran s 3 kolonami
    {
        izpis strani tipa s 3 kolonami z vsemi klici iz podatkovne baze
    }
?>

<?php
    če je tip spletne strani tipa podstran z 2 kolonami
    {
        izpis strani tipa podstran z 2 kolonami z vsemi klici iz podatkovne baze
    }
?>

<?php
    če je tip spletne strani tipa podstran z galerijo
    {
        izpis strani tipa podstran z galerijo z vsemi klici iz podatkovne baze
    }
?>

<?php
    če je tip spletne strani tipa kontakti
    {
        izpis strani tipa kontakti z vsemi klici iz podatkovne baze
    }
?>
Izpis noge

```

Slika 20: Psevdo koda integracije v statično HTML stran.

5.7.1 Razlaga kode PHP kreiranih funkcij iz dodatka C

V datoteki *funkcije.php*, katero vključimo v spletno stran imamo nanizane funkcije in procedure, ki jih uporabljamo pri integraciji CMS sistema.

- `getAtributiStrani`

Funkcija nam vrne vse attribute strani, katere id podamo v argumentu. Funkcija iz podatkovne baze naredi poizvedbo vseh atributov strani, kot tudi ostalih parametrov: naslov strani, tip strani, itd. Funkcijo uporabljamo pri izpisu spletne strani v HTML predlogo.

- `createFriendlyUrlFromId`

Funkcija nam kreira primeren zapis nadpovezav med stranmi. Funkcija prejme kot argument id strani in vrne nadpovezavo do te strani. Če ima stran, kateri želimo kreirati brskalniku prijazno nadpovezavo med atributi *seo_url* (atribut namenjen za optimizacijo spletnih strani,

ki spremeni url naslov strani), potem funkcija kreira nadpovezavo iz tega atributa, drugače kreira povezavo v obliki *index.php?id=5*, kjer je 5 id strani.

5.7.2 Razlaga kode PHP določitve tipov strani iz dodatka B

5.7.2.1 Določitev tipov podstrani v PHP kodi

Programiranja se bomo lotili najprej z razdelitvijo tipov podstrani na različne tipe strani, ki se bodo prikazovali obiskovalcem. Ker ne bomo uporabljali samo funkcije, ki so že vključene v jeziku PHP, ampak še nekaj ostalih funkcij, ki jih hranimo v datoteki *funkcije.php* moramo to datoteko vključiti ob nalaganju spletne strani. V datoteki *funkcije.php* imamo tudi povezavo do datoteke *database_connect.php*, ki odpre povezavo s podatkovno bazo v kateri hranimo podatke. PHP koda določitve tipov je prikazana na sliki 21.

```
<?php
include("funkcije.php");
if(isset($get['id']))
{
    $stran = getAtributiStrani($get['id']);
    if($stran['tipid'] == 6) //Podstran 3 kolone
    {
        $page_type = 3;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 7) //Podstran 2 kolone
    {
        $page_type = 4;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 8) //Podstran z galerijo
    {
        $page_type = 5;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 9) //Podstran kontakti
    {
        $page_type = 6;
        $page = getAtributiStrani($get['id']);
    }
}
else
{
    $page_type = 0;          //index - vstopna stran
}
?>
```

Slika 21: Določitev tipov strani v PHP kodi.

Z funkcijo `isset($get['id'])` preverimo ali ima stran nastavljen kakšen id v url naslovu, ter če ima preverimo če ima stran s tem id-jem kateri od tipov strani s tipid 6, 7, 8 ali 9, ter tej podstrani določimo `$page_type` na katerega se bomo sklicevali pri nalaganju strani. Če stran v url naslovu nima nobenega id-ja, potem je to vstopna stran in ji določimo `$page_type`.

5.7.2.2 Integracija menija v kodi PHP na spletni strani

Na sliki 22 je prikazana implementacija menija spletne strani, ki smo ga uporabili na naši spletni strani. Meni se prikazuje na vsaki strani in podstrani, tako da se mora ob nalaganju nove strani spreminjati akcija nadpovezav z oznako *active*, ki menijsko nadpovezavo obarva drugačne barve.

```

<ul class="menu">
  <li ><a href="index.php" class="<?phpif($page_type == 0) echo
"active";?>"><span>Domov</span></a></li>
  <?php
  if(isset($get['id']))
  {
    $stran = getAtributiStrani($get['id']);
    if($stran['parent'] == 2)
      $id_navigacije = $stran['stranid'];
    else
    {
      while($stran['parent'] >2)
      {
        $stran = getAtributiStrani($stran['parent']);
      }
      $id_navigacije = $stran['stranid'];
    }
  }
  else
  {
    $id_navigacije = -1;
    $query_navi = mysql_query("SELECT stranid, naslov FROM stran WHERE parent = 2 AND
deleted = 0 AND skrij = 0 ORDER BY red ASC");
    while($navi
= mysql_fetch_array($query_navi))
    {
      $link = createFriendlyUrlFromId($navi['stranid'], $navi['naslov']);
      if($navi['stranid'] == $id_navigacije &&isset($get['id']))
      {
        $active =
"active";
      }
      else
      {
        $active = "";
      }
      $query_podnavi = mysql_query("SELECT stranid, naslov FROM stran WHERE parent =
".$navi['stranid']." AND deleted = 0 AND skrij = 0 ORDER BY red ASC");

      echo "<li><a
href='{ $link }' class='{ $active }'><span>{ $navi['naslov'] }</span></a><ul>";

      while($podnavi = mysql_fetch_array($query_podnavi))
      {
        $link2 = createFriendlyUrlFromId($podnavi['stranid'],
$podnavi['naslov']);
        echo "<li><a href='{ $link2 }'>{ $podnavi['naslov'] }</a></li>";
      }
      echo "</ul></li>";
    }
  }
}
?>
</ul>

```

Slika 22: Implementacija menija v kodi PHP.

Pri implementaciji navigacijskega menija na spletni strani smo menijsko akcijo vstopne strani ohranili in je ni možno spreminjati znotraj CMS vmesnika. Pri tej menijski možnosti samo preverjamo, če se obiskovalec nahaja na vstopni strani in če se izpisujemo razred *active* znotraj nadpovezave, kar nam preko CSS datoteke drugače obarva povezavo.

Meni izpisujemo v dveh zankah, prva zanka je za izpis glavnega menija, druga vgnezdena zanka pa za izpis podmenija. Pri izpisih vsakič tudi preverjamo ali je obiskovalec na tej strani in, če je tej hiperpovezavi prav tako pripnemo element *active*.

5.7.2.3 Integracija v kodi PHP podstran tipa podstran z galerijo

Slika 23 prikazuje integracijo podstrani z galerijo, katero uporabniki polnijo preko CMS vmesnika v PHP jeziku, ki smo ga vključili v obstoječi HTML označevalni jezik.

```

<?php
if($page_type == 5) { //PODSTRAN Z GALERIJO ##### ?>
<div class="content padngbtm">
<div class="main_wrap ">
<div class="main_wrap_support topMargin">
<div id="content" class="margin_adjust">
<?php
    $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE
stranid = ".$get['id']." ORDER BY red ASC"));
    $podstranGal = getAtributiStrani($query['stranid']);
    ?>
    <h3
class="indexhead"><?=$podstranGal['naslov'];?></h3><p><?=$podstranGal['text'];?></p>
<ul class="img-gallery main-wrap">
<?php
$query="SELECT galerijaid, file, opis, red FROM galerija WHERE stranid =
".$get['id']." ORDER BY red ASC";
$result=mysql_query($query);
while ($line=mysql_fetch_array($result,MYSQL_ASSOC))
{ ?>
    <li><a href="vsebina/<?=$line['file'];?>" rel="prettyPhoto[gallery2]">"
/></a></li>
<? } ?>
</ul>
<!--END OF CONTENT AREA-->
</div>
</div>
</div>
</div>
<? } //KONEC PODSTRAN Z GALERIJO ##### ?>

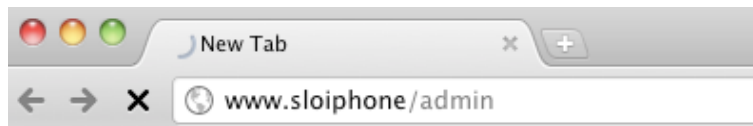
```

Slika 23: Integracija podstrani z galerijo v PHP kodi.

Za prikaz podstrani z galerijo smo morali najprej preveriti, če je stran, ki jo prikazujemo pravi tipa, ki smo ga določili že predčasno (`page_type`). V tem PHP bloku vnesemo elemente HTML-ja ter nadomestimo naslove, besedila, ki jih prikazujemo na spletni strani s poizvedbo iz podatkovne baze. Na tej podstrani je tudi prikazana implementacija galerije slik, katere hranimo na strežniku v mapi `vsebina`, pot do te datoteke pa v podatkovni bazi. Noga (ang. footer) podstrani se prikazuje že pred samim nalaganjem strani, tako da se v tem PHP bloku dejansko zamenjajo le podatki, ki se prikazujejo na tej podstrani.

5.8 UPORABA CMS SISTEMA

Uporabniki spletne strani z integriranim vmesnikom za urejanje spletne vsebine se v sistem prijavijo preko URL naslova v brskalniku. Imenu domene je potrebno dodati še ime mape do dostopa datotek sistema, v našem primeru admin, tako kot prikazuje slika 24.






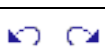
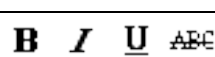

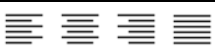

Slika 24: URL dostop do spletnega vmesnika CMS.



Uporabniki se potem s svojim uporabniškim imenom in geslom, ki jim ga je dodelil administrator prijavijo v sistem. Okolje spletnega vmesnika je za vse uporabnike podobno, le pri nekaterih je nekaj dodatnih omejitev. Ko je uporabnik prijavljen v sistem lahko začne z urejanjem spletne vsebine.

5.8.1 Urejanje vsebine z FCK editorjem

Urejevalnik nam omogoča veliko možnosti pri oblikovanju besedila, tako kot nam to omogočajo urejevalniki MS Word in podobni. Naprednejši uporabniki lahko vnašajo besedilo tudi v obliki izvorne kode, kar pomeni, da lahko oblikujejo spletno vsebino v HTML kodi. Prav tako je možno vezati določene HTML značke na CSS stilske predloge.

Večina običajnih uporabnikov uporablja funkcije FCKeditor-ja, ki so predstavljene v tabeli.

	Kopira označeno besedilo.
	Prilepi podatke in besedilo.
	Prilepi vsebino kopirano iz programa MS Word ali podobnih.
	Korak naprej in korak nazaj.
	Določi označenemu besedilu krepko, ležeče, podčrtano in prečrtano.
	Kreira naštevanje s številkami ali navadeno naštevanje.
	Poravnava besedila – levo, sredinsko, desno in centralno poravnavo.
	Ustvari tabelo.

	Vstavljanje slike.
	Spreminjanje barve besedila.

5.8.2 Dodajanje slik v galerijo

Na straneh, kjer je vključena galerija imajo uporabniki možnost dodajanja posameznih slik v galerijo. Uporabniku se odpre novo okno, kjer ima možnost dodajanja posameznih slik in vsaki sliki dodati opis, kot je prikazano na sliki 25.

Urejanje galerije

Dodaj novo sliko v galerijo

Gumb za dodajanje slike

Slika: Browse...

Naslov slike:

dodaj sliko

Trenutne slike v galeriji



poglej | uredi | X



poglej | uredi | X



poglej | uredi | X

Zapri okno

Slika 25: Dodajanje slik v galerijo v sistemu CMS.

Ko uporabniki dodajo slike, ki jih želijo uporabiti v galeriji imajo tudi možnost spreminjanja vrstnega reda slik v galeriji. Vrstni red lahko enostavno spremenijo s povleci in spusti možnostjo (ang. drag and drop), ter shranijo nov vrstni red.

6 SKLEP

Diplomska naloga prikazuje smernice, orodja, tehnologije in tehnike, ki jih razvijalec potrebuje pri integraciji spletnega vmesnika CMS v statično spletno stran. Tovrstni CMS sistem je možno vpeljati v katerokoli statično HTML stran kar omogoča ohranjanje predlog spletne strani v primeru, ko lastnik spletne strani želi zamenjati vmesnik pri tem pa se podoba spletne strani ne spremeni. V diplomski nalogi smo integrirali CMS sistem v statično predlogo spletne strani. Spletni vmesnik CMS je implementiran v nekaj spletnih straneh. Uporabniki tega sistema so s tovrstno rešitvijo zadovoljni, saj je enostaven za uporabo.

Ker je potreba po dinamičnih spletnih straneh vedno večja, saj je s tem povezana tudi optimizacija same spletne strani oziroma rangiranje v večjih iskalnikih, se vedno več ljudi in podjetij poslužuje tovrstnih sistemov.

Priprava delovnega okolja z namestitvijo paketa XAMPP s spletnim stražnikom Apache, sistemom za upravljanje s podatkovnimi bazami MySQL in programskim jezikom PHP je potekalo brez težav.

Pri integraciji vmesnika v statično spletno stran nismo naleteli na večje težave, ki jih ne bi mogli odpraviti. Ena od težav, ki se je pojavila pri testiranju vmesnika na Microsoftovem IIS strežniškem sistemu je, da mora biti PHP koda napisana med pravičnimi značkami (`<?php ?>`), kar Apachejevi strežniki tolerirajo (`<? ?>`). Nekaj težav se je pojavilo tudi pri preoblikovanju statične predloge spletne strani v obliko, ki je bila v zahtevah naročnika, predvsem s CSS stilskimi predlogami, vendar smo jih ob pomoči Firebug dodatka spletnega brskalnika Firefox hitro odpravili.

Možnosti za razširitev tovrstnega sistema je še veliko, saj omogoča nekaj osnovnejših funkcij, ki zadovoljujejo za večino predstavitev spletnih strani. Vendar vsaka nova funkcionalnost uporabnike obremeni in s tem sistem postaja vedno bolj nepregleden.

7 VIRI IN LITERATURA

- [1] B. Boiko: Content Management Bible, Metatorial Services Inc. & Hungry Minds Inc., New York, 2002
- [2] P. Browning, M. Lowndes: Content Management Systems, JISC TechWatch Report, University of Bristol, 2001
- [3] Cristian Darie, Bogdan Brinzarea, Filip Cherches, Mihai Bucica. Ajax and PHP – building Responsive Web Aplicatons. Založba PACKT, 2006
- [4] Marc Delisle: Mastering phpMyAdmin 3.3.x for Effective MySQL Management. Založba Packt Publishing
- [5] (2011) FCKeditor. Dostopno na:
<http://drupal.org/project/fckeditor>
- [6] (2011) Firebug Web Development Evolved. Dostopno na:
<http://getfirebug.com/>
- [7] (2011) Firefox web browser. Dostopno na:
<http://www.mozilla.com/sl/firefox/features/>
- [8] (2011) HTML & CSS. Dostopno na:
<http://www.w3.org/standards/webdesign/htmlcss>
- [9] (2011) JavaScript Introduction. Dostopno na:
http://www.w3schools.com/js/js_intro.asp
- [10] (2011) TextMate. Dostopno na:
<http://macromates.com/>
- [11] (2011) What is MySQL. Dostopno na:
<http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>
- [12] (2011) What is PHP. Dostopno na:
<http://www.amazon.com/PHP5-MySQL-Bible-Tim-Converse/dp/0764557467#reader>
- [13] (2011) XAMPP. Dostopno na:
<http://www.ibm.com/developerworks/linux/library/l-xampp/>

DODATEK A

Izvorna koda statične spletne strani

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>TemplateWorld | Times Arena</title>
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
<link href="css/contentslider.css" rel="stylesheet" type="text/css" media="all" />
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script src="js/contentslider.js" type="text/javascript"></script>
<!--SCRIPT FOR MENU-->
<script type="text/javascript">
<!--
$(document).ready(function () {

    $('.menu li').hover(
        function () {
            //show its submenu
            $('ul', this).fadeIn();

        },
        function () {
            //hide its submenu
            $('ul', this).fadeOut();

        }
    );

});
-->
</script>
<script type="text/javascript" src="js/easySlider1.5.js" ></script>
<script type="text/javascript">
    $(document).ready(function(){
        $('#slider').easySlider({
            auto: true,
            continuous: true
        });
    });
</script>
<link href="fonts/stylesheet.css" rel="stylesheet" type="text/css" media="all" />
<script type="text/javascript" src="js/contactable/jquery.validate.pack.js"></script>
<script type="text/javascript" src="js/contactable/jquery.contactable.js"></script>
<link rel="stylesheet" href="js/contactable/contactable.css" type="text/css" />
<script type="text/javascript">
    $(function(){
        $('#contact').contactable({
            subject: 'A Feedback Message'
        });
    });
</script>

<!--[if IE 8]>
<style type="text/css">
.table_wrap p.heading_3 {width:449px;}
</style>
<![endif]-->
<!--[if IE 7]>
<style type="text/css">
#contactForm{height:365px;}
form#contactForm .submit{margin-top:0px;}
.center text img.shadow{height:0px}
</style>
<![endif]-->
</head>

<body>
<div id="contact"> </div>
<div id="fullwidth">
    <div class="main_wrap">
        <div class="main_wrap_support"> <a href="index.html" class="logo"></a>
            <ul class="menu">
                <li ><a href="index.html" class="active"><span>Home</span></a>
                    <ul>
                        <li><a href="home.html" class="noborder">Static</a></li>
                    </ul>
                </li>
                <li><a href="#" ><span> Options</span></a>
                    <ul>
                        <li><a href="short-codes.html">Short Codes</a></li>
                        <li><a href="left-panel.html">Left Panel</a></li>
                        <li><a href="right-panel.html">Right Panel</a></li>
                    </ul>
                </li>
            </ul>
        </div>
    </div>
</div>

```

```

        <li><a href="3-columns.html">3 Columns</a></li>
        <li><a href="misc.html">Miscellaneous</a></li>
        <li><a href="scroll.html">Scroll Page</a></li>
        <li><a href="dynamic-tabs.html" class="noborder">Dynamic Tabs</a></li>
    </ul>
</li>
<li><a href="#"><span>Gallery</span></a>
    <ul>
        <li><a href="gallery1.html">Gallery 1</a></li>
        <li><a href="gallery2.html">Gallery 2</a></li>
        <li><a href="gallery3.html" class="noborder">Gallery 3</a></li>
    </ul>
</li>
<li><a href="#"><span>Blog</span></a>
    <ul>
        <li><a href="blog.html">Blog Post</a></li>
        <li><a href="blog-innerpage.html" class="noborder">Full Post</a></li>
    </ul>
</li>
<li><a href="contact.html"><span>Contact</span></a></li>
</ul>
<div class="bannerwrap">
    <div class="bannerwraplft">
        <h2>Universal Website </h2>
        <p>Template Name' is a multi-purpose template
        created by experienced team at TemplateWorld.</p>
        <a href="#" class="viewdemo"></a> </div>
    <div class="bannerwraprht">
        <div id="slidePanel">
            <div id="slider1" class="sliderwrapper">
                <div class="contentdiv">  </div>
                <div class="contentdiv">  </div>
                <div class="contentdiv">  </div>
                <div id="paginate-slider1" class="pagination"></div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
<div class="content">
    <div class="main_wrap">
        <div class="main_wrap_support">
            <div class="mini_wrap">
                <h3 class="indexhead">What we do</h3>
                <p class="indexpara">Template Name' is a multi-purpose template created by experienced team at
                TemplateWorld, that will fit most business and personal website needs at a very competitive price, and top-
                notch quality. </p>
                <p class="indexpara">You will need minimal knowledge yet you can easily customize it to the
                best.website needs at a very competitive price, and top-notch quality.</p>
            </div>
            <div class="mini_wrap">
                <h3 class="indexhead">Latest Project</h3>
                
                <p class="indexpara">'Template Name' is a multi-purpose template created by experienced team at
                TemplateWorld, that will fit most business and personal </p>
            </div>
            <div class="mini_wrap">
                <h3 class="indexhead">Who We Are</h3>
                <p class="indexpara">Template Name' is a multi-purpose template created by experienced team at
                TemplateWorld, that will fit most business and personal website needs at a very competitive price, and top-
                notch quality. </p>
                <p class="indexpara">You will need minimal knowledge yet you can easily customize it to the
                best.website needs at a very competitive price, and top-notch quality.</p>
            </div>
        </div>
    </div>
</div>
</div>
<div class="centertext">
    <div class="main_wrap">
        <div class="main_wrap_support"> 
        <div class="two-equals">
            <div class="indextwo">
                <h4 class="headtwoeq1"> Web
                Development </h4>
                 </div>
                <p class="paratwoeq1">'Template Name' is a multi-purpose template created by experienced team at
                TemplateWorld, that will fit most business and personal website needs at a very competitive price, and top-
                notch quality. You will need minimal knowledge yet you can easily customize it to the best of its
                potential. You will need minimal knowledge yet you can.</p>
            </div>
            <div class="two-equals">
                <p class="paratwoeq1">'Template Name' is a multi-purpose template created by experienced team at
                TemplateWorld, that will fit most business and personal website needs at a very competitive price, and top-
                notch quality. You will need minimal knowledge yet you can easily customize it to the best of its
                potential. You will need minimal knowledge yet you can.</p>
            </div>
            <div class="indextwo">
                <h4 class="headtwoeq12">CMS<br />
                Integration</h4>
            </div>
        </div>
    </div>
</div>

```

```

         </div>
</div>
<h5 class="indexhead">Client<br />
Work</h5>
<div class="sliderwrapindex">
  <div id="container">
    <div id="slider">
      <ul>
        <li>       </li>
        <li>       </li>
        <li>       </li>
        <li>       </li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
<div class="btmcontent">
  <div class="main_wrap">
    <div class="main_wrap_support">
      <div class="left_pannel">
        <p class="testimoli">Proin fringilla, diam in placerat iaculis,
erat velit hendrerit nulla, ut commodo urna libero vitae sem. Donec l erat velit hendrorem dolor,
condimentum et sillicitudin eu, facilisis vitae erat.....ollicitudin eu, facilisis vitae erat.....</p>
      </div>
      <div class="mini_wrap">
        <h4 class="minihead">Support</h4>
        <p class="minipara">Lorem ipsum dolor sit amet, consectetur adipi.</p>
        <h4 class="minihead lessmatop">Address:</h4>
        <p class="minipara">1090 25th Place, Bradenton, FL 34203,<br />
United States.
Call: (001) 234-567-890</p>
        <h4 class="minihead lessmatop">Cross Browser Compatible</h4>
         </div>
      </div>
    </div>
  </div>
</div>
<div id="footerbg">
  <div class="main_wrap">
    <div class="main_wrap_support">
      <p> All Rights reserved. Designed by: <a href="http://www.templateworld.com" target="_blank"
class="temp">Template World</a></p>
      <ul class="nav2">
        <li><a href="#">Home</a></li>
        <li class="sep">|</li>
        <li><a href="#">Options</a></li>
        <li class="sep">|</li>
        <li><a href="#">Gallery</a></li>
        <li class="sep">|</li>
        <li><a href="#">Blog</a></li>
        <li class="sep">|</li>
        <li><a href="#">Contact</a></li>
      </ul>
    </div>
  </div>
</div>
</div>
<script type="text/javascript">
featureddcontentslider.init({
  id: "slider1", //id of main slider DIV
  contentsource: ["inline", ""], //Valid values: ["inline", ""] or ["ajax", "path_to_file"]
  toc: "#increment", //Valid values: "#increment", "markup", ["label1", "label2", etc]
  nextprev: ["", ""], //labels for "prev" and "next" links. Set to "" to hide.
  revealttype: "click", //Behavior of pagination links to reveal the slides: "click" or "mouseover"
  enablefade: [true, 0.2], //[true/false, fadegree]
  autorotate: [true, 3000], //[true/false, pausetime]
  onChange: function(previndex, curindex){ //event handler fired whenever script changes slide
    //previndex holds index of last slide viewed b4 current (1=1st slide, 2nd=2nd etc)
    //curindex holds index of currently shown slide (1=1st slide, 2nd=2nd etc)
  }
})
</script>
</body>
</html>

```

DODATEK B

Izvorna koda spletne strani s CMS sistemom

```

<?php
//dostop do phpmyadmina      SITE: SLO IPHONE
include("funkcije.php");

if(isset($get['id']))
{
    $stran = getAtributiStrani($get['id']);
    if($stran['tipid'] == 6) //Podstran 3 kolone
    {
        $page_type = 3;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 7) //Podstran 2 kolone
    {
        $page_type = 4;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 8) //Podstran z galerijo
    {
        $page_type = 5;
        $page = getAtributiStrani($get['id']);
    }
    else if($stran['tipid'] == 9) //Podstran kontakti
    {
        $page_type = 6;
        $page = getAtributiStrani($get['id']);
    }
}
else
{
    $page_type = 0; //index - vstopna stran
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>iPhone 4 v Sloveniji</title>
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
<link rel="stylesheet" href="css/validationEngine.jquery.css" type="text/css" media="screen" title="no
title" charset="utf-8" />
<!-- END of Css file for form validation popups -->
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
<link href="css/contentslider.css" rel="stylesheet" type="text/css" media="all" />
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<!-- Js file for form validation popups -->
<script src="js/jquery.validationEngine-en.js" type="text/javascript"></script>
<script src="js/jquery.validationEngine.js" type="text/javascript"></script>
<!-- End of Js file for form validation popups -->
<!--*****-->
<script src="js/contentslider.js" type="text/javascript" ></script>
<!--SCRIPT FOR MENU-->
<script type="text/javascript">
<!--
$(document).ready(function () {

    $('.menu li').hover(
        function () {
            //show its submenu
            $('ul', this).fadeIn();
        },
        function () {
            //hide its submenu
            $('ul', this).fadeOut();
        }
    );
    // For form validation
    $("#formID").validationEngine({
        success : function() {
            $('#load').show();
            var dataString = 'name='+ $("input[name]").val() + '&email=' + $("input[email]").val() +
            '&comments=' + $('textare#comments').val();
            $.ajax({
                type: "POST",
                url: "bin/process.php",
                data: dataString,
                success: function(msg) {
                    $('#load').hide();
                    $('#message').html(msg)
                    .hide()

```

```

        .fadeIn(1500, function() {
            $('#message').append("");
        });
        $("input#name").val('Name');
        $("input#email").val('E-mail');
        $('textarea#comments').val('Message');
    }
    });
}
})
// End of form Validation

});
-->
</script>
<script type="text/javascript" src="js/easySlider1.5.js" ></script>
<script type="text/javascript">
    $(document).ready(function(){
        $('#slider').easySlider({
            auto: true,
            continuous: true
        });
    });
</script>
<link href="fonts/stylesheet.css" rel="stylesheet" type="text/css" media="all" />
<script type="text/javascript" src="js/contactable/jquery.validate.pack.js"></script>
<script type="text/javascript" src="js/contactable/jquery.contactable.js"></script>
<link rel="stylesheet" href="js/contactable/contactable.css" type="text/css" />
<script type="text/javascript">
    $(function(){
        $('#contact').contactable({
            subject: 'A Feeback Message'
        });
    });
</script>

<!--[if IE 8]>
<style type="text/css">
    .table_wrap p.heading_3 {width:449px;}
</style>
<![endif]-->
<!--[if IE 7]>
<style type="text/css">
#contactForm{height:365px;}
form#contactForm .submit{margin-top:0px;}
.centertext img.shadow{height:0px}
</style>
<![endif]-->
<script src="js/jquery.js" type="text/javascript" charset="utf-8"></script>
<link rel="stylesheet" href="css/prettyPhoto.css" type="text/css" media="screen" charset="utf-8" />
<script src="js/jquery.prettyPhoto.js" type="text/javascript" charset="utf-8"></script>
</head>

<body>

<div id="fullwidth">
    <div class="main_wrap">
        <div class="main_wrap_support"> <a href="index.php" class="logo"></a>
        <ul class="menu">
            <li ><a href="index.php" class="<? if($page_type == 0) echo
"active";?>"><span>Domov</span></a></li>
            <?php
                if(isset($get[id]))
                {
                    $stran = getAtributiStrani($get['id']);
                    if($stran['parent'] == 2)
                        $id_navigacije = $stran['stranid'];
                    else
                    {
                        while($stran['parent'] > 2)
                        {
                            $stran = getAtributiStrani($stran['parent']);
                        }
                        $id_navigacije = $stran['stranid'];
                    }
                }
                else
                    $id_navigacije = -1;

                $query_navi = mysql_query("SELECT stranid, naslov FROM stran WHERE parent = 2 AND
deleted = 0 AND skrij = 0 ORDER BY red ASC");
                while($navi = mysql_fetch_array($query_navi))
                {
                    $link = createFriendlyUrlFromId($navi['stranid'], $navi['naslov']);
                    if($navi['stranid'] == $id_navigacije && isset($get['id']))
                    {
                        $active = "active";
                    }
                }
            }
        }
    }

```

```

        else
            $active = "";
            $query_podnavi = mysql_query("SELECT stranid, naslov FROM stran WHERE parent =
".$navi['stranid']."' AND deleted = 0 AND skrij = 0 ORDER BY red ASC");
            echo "<li><a href='{ $link}'
class='{ $active}'><span>{$navi['naslov']}</span></a><ul>";
                while($podnavi = mysql_fetch_array($query_podnavi))
                {
                    $link2 = createFriendlyUrlFromId($podnavi['stranid'], $podnavi['naslov']);
                    echo "<li><a href='{ $link2}'>{$podnavi['naslov']}</a></li>";
                }
            echo "</ul></li>";
        }?>
    </ul>
    <? if($page_type == 0) { //INDEX #####?>
    <div class="bannerwrap">
        <div class="bannerwraplft">
            <?php
                $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 4 ORDER
BY red ASC"));
                $index = getAtributiStrani($query['stranid']);
            ?>
            <h2><?=$index['naslov'];?></h2>
            <p><?=$index['text-left'];?></p>
        </div>
        <div class="bannerwraprht">
            <div id="slidePanel">
                <div id="slider1" class="sliderwrapper">
                    <?php
                        $query="SELECT galerijaid, file, opis, red FROM galerija WHERE stranid = 4 ORDER BY
red ASC";
                        $result=mysql_query($query);
                        while ($line=mysql_fetch_array($result,MYSQL_ASSOC))
                        { ?>
                            <div class="contentdiv">  </div>
                            <? } ?>
                        <div id="paginate-slider1" class="pagination"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="content">
        <div class="main_wrap">
            <div class="main_wrap_support">
                <div class="mini_wrap">
                    <?php
                        $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 5 ORDER
BY red ASC"));
                        $index = getAtributiStrani($query['stranid']);
                    ?>
                    <h3 class="indexhead"><?=$index['naslov'];?></h3>
                    <p class="indexpara"><?=$index['text'];?></p>
                </div>
                <div class="mini_wrap">
                    <?php
                        $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 7 ORDER
BY red ASC"));
                        $index = getAtributiStrani($query['stranid']);
                    ?>
                    <h3 class="indexhead"><?=$index['naslov'];?></h3>
                    <p class="indexpara"><?=$index['text'];?></p>
                </div>
                <div class="mini_wrap">
                    <?php
                        $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 6 ORDER
BY red ASC"));
                        $index = getAtributiStrani($query['stranid']);
                    ?>
                    <h3 class="indexhead"><?=$index['naslov'];?></h3>
                    
                    <p class="indexpara"><?=$index['text'];?></p>
                </div>
            </div>
        </div>
    </div>
    <div class="centertext">
        <div class="main_wrap">
            <div class="main_wrap_support"> 
            <div class="two-equals topMargin">
                <?php
                    $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 8 ORDER
BY red ASC"));
                    $index = getAtributiStrani($query['stranid']);
                ?>
                <h3 class="indexhead"><?=$index['naslov'];?></h3>
                <p class="main"><?=$index['text'];?></p>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="two-equals topMargin">
        <?php
            $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 9 ORDER
BY red ASC"));
            $index = getAtributiStrani($query['stranid']);
            ?>
                <h3 class="indexhead"><?=$index['naslov'];?></h3>
                <p class="main"><?=$index['text'];?></p>
            </div>
            <?php
                $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 10 ORDER
BY red ASC"));
                $index = getAtributiStrani($query['stranid']);
                ?>
                    <h5 class="indexhead"><?=$index['naslov'];?></h5>
                    <div class="sliderwrapindex">
                    <div id="container">
                        <div id="slider">
                            <ul>
                                <li>
                                    <?php
                                        $query="SELECT galerijaid, file, opis, red FROM galerija WHERE stranid = 10 ORDER BY
red ASC";
                                        $result=mysql_query($query);
                                        while ($line=mysql_fetch_array($result,MYSQL_ASSOC))
                                        { ?>
                                            
                                        <? } ?>
                                        </li></ul>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="btmcontent">
                    <div class="main_wrap">
                        <div class="main_wrap_support">
                        <div class="left_pannel">
                            <p class="testimoli">
                            <?
                                $query_stranke = mysql_query("SELECT stranid, naslov FROM stran WHERE parent = 3 ORDER BY
RAND() LIMIT 1");
                                while($izjave = mysql_fetch_array($query_stranke))
                                {
                                    $izjava = getAtributiStrani($izjave['stranid']);
                                    $text = $izjava['izjava'];
                                } ?>
                                <?=$text;?><br /><?=$izjava['naslov'];?></p>
                            </div>
                        <div class="mini_wrap">
                            <?php
                                $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 14 ORDER
BY red ASC"));
                                $index = getAtributiStrani($query['stranid']);
                                ?>
                                    <h4 class="minihead"><?=$index['podpora'];?></h4>
                                    <p class="minipara"><?=$index['podpora_text'];?></p>
                                    <h4 class="minihead lessmatop"><?=$index['naslov'];?></h4>
                                    <p class="minipara"><?=$index['naziv'];?><br />
                                    <?=$index['ulica'];?><br />
                                    <?=$index['mesto'];?><br />
                                    <?=$index['drzava'];?><br />
                                    GSM: <?=$index['gsm'];?><br />
                                    TEL: <?=$index['tel'];?><br />
                                    FAX: <?=$index['fax'];?><br />
                                    E-mail: <a href="mailto:<?=$index['mail'];?>"><?=$index['mail'];?></a><br /></p>
                                </div>
                            </div>
                        </div>
                    <? } //KONEC INDEX ##### ?>

<? if($page_type == 5) { //PODSTRAN Z GALERIJO ##### ?>
<div class="content padngbtm">
    <div class="main_wrap ">
        <div class="main_wrap_support topMargin">
            <div id="content" class="margin_adjust">
                <?php
                    $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid =
".$get['id']." ORDER BY red ASC"));
                    $podstranGal = getAtributiStrani($query['stranid']);
                    ?>
                        <h3 class="indexhead"><?=$podstranGal['naslov'];?></h3>
                        <p><?=$podstranGal['text'];?></p>
                        <ul class="Img-gallery main-wrap">
                            <?php
                                $query="SELECT galerijaid, file, opis, red FROM galerija WHERE stranid = ".$get['id']." ORDER
BY red ASC";

```

```

        $result=mysql_query($query);
        while ($line=mysql_fetch_array($result,MYSQL_ASSOC))
        { ??
            <li><a href="vsebina/<?=$line['file'];?>" rel="prettyPhoto[gallery2]">" /></a></li>

            <? } ??
        </ul>
        <!--END OF CONTENT AREA-->
    </div>
</div>
</div>
<? } //PODSTRAN Z GALERIJO ##### ?>
<? if($page_type == 4) { //PODSTRAN 2 KOLONI ##### ?>
<div class="content padngbtm">
    <div class="main_wrap">
        <div class="main_wrap_support topMargin">
            <div id="content" class="margin_adjust">
                <div class="two-equals topMargin">
                    <?php
                        $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid =
.$get['id']. ORDER BY red ASC"));
                        $podstran2 = getAtributiStrani($query['stranid']);
                    ?>
                    <h3 class="indexhead"><?=$podstran2['naslov'];?></h3>
                    <p class="main"><?=$podstran2['text-left'];?></p>
                    </div>
                    <div class="two-equals topMargin">
                        <h3 class="indexhead"><?=$podstran2['naslov-desno'];?></h3>
                        <p class="main"><?=$podstran2['text-right'];?></p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<? } //KONEC PODSTRAN 2 KOLONI ##### ?>
<? if($page_type == 3) { //PODSTRAN 3 KOLONE ##### ?>

    <?php
        $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid =
.$get['id']. ORDER BY red ASC"));
        $podstran3 = getAtributiStrani($query['stranid']);
    ?>
    </div>
</div>
<div class="content">
    <div class="main_wrap">
        <div class="main_wrap_support">
            <div class="mini_wrap">
                <p class="indexpara"><?=$podstran3['text-left'];?></p>
            </div>
            <div class="mini_wrap">
                <p class="indexpara"><?=$podstran3['text-center'];?></p>
            </div>
            <div class="mini_wrap">
                <p class="indexpara"><?=$podstran3['text-right'];?></p>
            </div>
        </div>
    </div>
</div>

<? } //KONEC PODSTRAN 3 KOLONE ##### ?>

<? if($page_type == 6) { //KONTAKT ##### ?>
<div class="content padngbtm">
    <div class="main_wrap ">
        <div class="main_wrap_support topMargin">
            <div id="content" class="margin_adjust">
                <div class="left_annel">
                    <h3 class="indexhead">Kontaktirajte nas!</h3>
                    <form action="#" class="mail" id="formID">
                        <input name="name" type="text" value="Name" id="name"
class="validate[required,exemptString[Name]]" onfocus="if(this.value=='Name')this.value='';"
onblur="if(this.value=='')this.value='Name';" />
                        <input name="email" type="text" value="E-mail" id="email"
class="validate[required,custom[email],exemptString[E-mail]]" onfocus="if(this.value=='E-
mail')this.value='';" onblur="if(this.value=='')this.value='E-mail';" />
                        <textarea name="comments" cols="2" rows="5" class="validate[required,exemptString[Message]]"
id="comments" onfocus="if(this.value=='Message')this.value='';"
onblur="if(this.value=='')this.value='Message';">Message</textarea>
                        <input type="submit" name="submit" value="Pošlji" class="more_button" />
                    <!-- Change the loading style after Submit -->
                    <span id="load" style="float:left; margin-top:30px; margin-left:10px; display:none; font-
family:Arial, Helvetica, sans-serif; font-size:12px; color:#CCCCCC;" > Please wait ...</span>
                    <!-- End of loading style -->
                    <!-- DIV where the successful form submit message displayed -->
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <div id="message" style="display:none; float:left; color:#000; font-family:Arial, Helvetica,
sans-serif;"></div>
        <!-- End Of div -->
    </form>
</div>
<?php
    $query = mysql_fetch_array(mysql_query("SELECT stranid, naslov FROM stran WHERE stranid = 14
ORDER BY red ASC"));
    $kontakti = getAtributiStrani($query['stranid']);
    ?>
<div class="mini_wrap">

    <h3 class="indexhead">Informacije</h3>
    <p class="main"><b><?=$kontakti['naziv'];?></b><br /><?=$kontakti['ulica'];?><br />
    <?=$kontakti['mesto'];?><br />
    <?=$kontakti['drzava'];?><br />
    <b>GSM:</b> <?=$kontakti['gsm'];?><br />
    <b>TEL:</b> <?=$kontakti['tel'];?><br />
    <b>FAX:</b> <?=$kontakti['fax'];?><br />
    <b>Mail:</b> <?=$kontakti['mail'];?><br />
    <br />
    </p>
    <h3 class="indexhead">Zemljevid</h3>
    <iframe class="map" width="300" height="190" frameborder="0" scrolling="no" marginheight="0"
marginwidth="0"
src="http://maps.google.com/maps?f=q&source=s_q&hl=en&geocode=&q=1090+25th+Place,+Bradenton
,+FL+34203,+United+States&ie=UTF8&hq=&hnear=1090+25th+St+W,+Bradenton,+Manatee,+Florida+34205&a
mp;ll=27.490152,-82.585945&spn=0.015989,0.025663&z=14&iwloc=A&output=embed"></iframe>
    <br />
    <small><a
href="http://maps.google.com/maps?f=q&source=embed&hl=en&geocode=&q=1090+25th+Place,+Braden
ton,+FL+34203,+United+States&ie=UTF8&hq=&hnear=1090+25th+St+W,+Bradenton,+Manatee,+Florida+3420
5&ll=27.490152,-82.585945&spn=0.015989,0.025663&z=14&iwloc=A" class="more">Većji
pogled</a></small> </div>
    </div>
</div>
</div>
        <!--END OF CONTENT AREA-->
<? } //KONEC KONTAKTI ##### ?>
    </div>
    </div>
</div>
</div>

<div id="footerbg">
    <div class="main_wrap">
        <div class="main_wrap_support">
            <p> &copy; Copyright: <a href="#" target="_blank" class="temp">Jure Čebašel</a></p>
            <ul class="nav2">
                <li><a href="#">Home</a></li>
            </ul>
            <?php
                if(isset($get[id]))
                {
                    $stran = getAtributiStrani($get[id]);
                    if($stran['parent'] == 2)
                        $id_navigacije = $stran['stranid'];
                    else
                    {
                        while($stran['parent'] > 2)
                        {
                            $stran = getAtributiStrani($stran['parent']);
                        }
                        $id_navigacije = $stran['stranid'];
                    }
                }
                else
                {
                    $id_navigacije = -1;
                    $sep="sep";
                    $query_navi = mysql_query("SELECT stranid, naslov FROM stran WHERE parent = 2 AND
deleted = 0 AND skrij = 0 ORDER BY red ASC");
                    while($navi = mysql_fetch_array($query_navi))
                    {
                        $link = createFriendlyUrlFromId($navi['stranid'], $navi['naslov']);
                        echo "<li class='{ $sep }'></li><li><a href='{ $link }'>{ $navi['naslov'] }</a></li>";
                    }
                }
            ?>
        </ul>
    </div>
</div>
</div>

<script type="text/javascript">
featuredcontentslider.init({
    id: "slider1", //id of main slider DIV
    contentsource: ["inline", ""], //Valid values: ["inline", ""] or ["ajax", "path_to_file"]
    toc: "#increment", //Valid values: "#increment", "markup", ["label1", "label2", etc]
    nextprev: ["", ""], //labels for "prev" and "next" links. Set to "" to hide.

```

```
    revealtype: "click", //Behavior of pagination links to reveal the slides: "click" or "mouseover"
    enablefade: [true, 0.2], //[true/false, fadedegree]
    autorotate: [true, 3000], //[true/false, pausetime]
    onChange: function(previndex, curindex){ //event handler fired whenever script changes slide
        //previndex holds index of last slide viewed b4 current (1=1st slide, 2nd=2nd etc)
        //curindex holds index of currently shown slide (1=1st slide, 2nd=2nd etc)
    }
})
</script>
<script type="text/javascript" charset="utf-8">
    $(document).ready(function(){
        $("a[rel^='prettyPhoto']").prettyPhoto();
    });
</script>
</body>
</html>
```

DODATEK C

Izvorna koda funkcij

```

<?
session_start();

include ("database_connect.php");

include ("escape.php");

function getAtributiStrani($stranid)
{
    @l;

    $query = "SELECT stranid, naslov, tipid, skrij, parent, sellable FROM stran WHERE stranid = " .
    $stranid;

    $result = mysql_query($query);

    $line = mysql_fetch_array($result, MYSQL_ASSOC);

    $l['stranid'] = $line['stranid'];

    $l['naslov'] = $line['naslov'];

    $l['skrij'] = $line['skrij'];

    $l['tipid'] = $line['tipid'];

    $l['parent'] = $line['parent'];

    $l['sellable'] = $line['sellable'];

    $query2 = "SELECT * FROM atribut WHERE deleted = 0 AND tipid = " . $line['tipid'] . " ORDER BY red
    ASC";

    $result2 = mysql_query($query2);

    while ($line2 = mysql_fetch_array($result2, MYSQL_ASSOC))
    {
        if ($line2['vrsta'] == 0) /* gre za TEKSTOVNI PODATEK */

            $l[$line2['spremenljivka']] = getTekstovniPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 10) /* gre za TEKSTOVNI PODATEK */

            $l[$line2['spremenljivka']] = getTekstovniPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 1) /* gre za DATUM */

            $l[$line2['spremenljivka']] = getDatumskiPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 2) /* gre za VSEBINSKI PODATEK - WYSIWYG (fckeditor) */

            $l[$line2['spremenljivka']] = getVsebinskiPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 3) /* gre za VSEBINSKI PODATEK - obiàajen textbox */

            $l[$line2['spremenljivka']] = getVsebinskiPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 4) /* gre za SLIKO */

            $l[$line2['spremenljivka']] = getTekstovniPodatek($line2['atributid'], $line['stranid']);

        if ($l[$line2['spremenljivka']] == "")

```

```

        $l[$line2['spremenljivka']] = "";

        /* if (!file_exists("../vsebina/".$l[$line2['spremenljivka']])) {
        $l[$line2['spremenljivka']]="ni_slike.gif"; } */

        if ($line2['vrsta'] == 5) /* gre za DATOTEKO */

            $l[$line2['spremenljivka']] = getTekstovniPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 12)

            $l[$line2['spremenljivka']] = getStevilskiPodatek($line2['atributid'], $line['stranid']);

        if ($line2['vrsta'] == 6) /* gre za GALERIJO */

            $l[$line2['spremenljivka']] = $line2['atributid'];

        if ($line2['vrsta'] == 7) /* gre za ANKETO */

            $l[$line2['spremenljivka']] = $line2['atributid'];

        if ($line2['vrsta'] == 8) /* gre za FAQ */

            $l[$line2['spremenljivka']] = $line2['atributid'];

        if ($line2['vrsta'] == 9) /* gre za FORMULAR */

            $l[$line2['spremenljivka']] = $line2['atributid'];

    }

    return $l;
}

function createFriendlyUrl($string)
{
    $before = array("°", " ", ".", ",", "\'", "!", "&", "š", "đ", "č", "ć", "ž", "ö", "š", "đ", "č", "ć", "ž", "ö", "ä", "ä", ":", "%");
    $after = array("", "-", "", "", "", "", "", "s", "d", "c", "c", "z", "o", "s", "d", "c", "c", "z", "o", "a", "a", "", "");

    return strtolower(str_replace($before, $after, $string));
}

function createFriendlyUrlFromId($id, $backup)
{
    $query_seo = mysql_query("SELECT tvrednost, atributid FROM podatek WHERE stranid = {$id} AND atributid IN(SELECT atributid FROM atribut WHERE spremenljivka = 'seo_url' AND deleted = 0)");

    if (mysql_affected_rows() > 0)
    {
        $row = mysql_fetch_array($query_seo);

        if ($row['atributid'] == 4)
        {
            $stran = getAtributiStrani($id);

            if ($stran['direct'] != "")
            {
                $url = str_replace("http://", "", $stran['direct']);

                $friendly_url = "http://" . $url;
            }
        }
    }
}

```

```
else if ($stran['id_strani'] != "")
    $friendly_url = createFriendlyUrlFromId($stran['id_strani'], $backup);
else if ($stran['ime_strani'] != "")
{
    if (file_exists("./" . $stran['ime_strani'] . ".php"))
        if ($stran['seo_url'] != "")
            $friendly_url = $stran['seo_url'];
        else
            $friendly_url = "index.php?page=" . $stran['ime_strani'];
        else
            $friendly_url = "";
    }
else
    $friendly_url = "";
}
else if ($row['tvrednost'] != "")
    $friendly_url = $id . "/" . createFriendlyUrl($row['tvrednost']);
else
    //$friendly_url = createFriendlyUrl($backup);
    $friendly_url = "index.php?id={$id}";
}
else
    $friendly_url = "index.php?id={$id}";

return $friendly_url;
//return "index.php?id=".$id; //to ti naredi navaden link
}
```

DODATEK D

SQL za kreiranje podatkovne baze

```

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Database: `empty`
-- Table structure for table `atribut`
--

CREATE TABLE IF NOT EXISTS `atribut` (
  `atributid` int(11) NOT NULL AUTO_INCREMENT,
  `tipid` int(11) NOT NULL DEFAULT '0',
  `opis` varchar(255) DEFAULT NULL,
  `vrsta` int(11) NOT NULL DEFAULT '0',
  `red` int(11) NOT NULL DEFAULT '0',
  `deleted` int(11) NOT NULL DEFAULT '0',
  `spremenljivka` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`atributid`),
  KEY `tipid` (`tipid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `atribut`
--
-----

--
-- Table structure for table `enovice`
--

CREATE TABLE IF NOT EXISTS `enovice` (
  `enoviceid` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(255) DEFAULT NULL,
  `prijavljen` int(11) NOT NULL DEFAULT '0',
  `ime` varchar(255) NOT NULL DEFAULT '',
  `priimek` varchar(255) NOT NULL DEFAULT '',
  `naslov` varchar(255) NOT NULL DEFAULT '',
  `posta` varchar(255) NOT NULL DEFAULT '',
  `kraj` varchar(255) NOT NULL DEFAULT '',
  `koncni` varchar(255) NOT NULL DEFAULT '',
  `arhitekt` varchar(255) NOT NULL DEFAULT '',
  `poslovni` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`enoviceid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `enovice`
--
-----

--
-- Table structure for table `galerija`
--

CREATE TABLE IF NOT EXISTS `galerija` (
  `galerijaid` int(11) NOT NULL AUTO_INCREMENT,
  `atributid` int(11) NOT NULL DEFAULT '0',
  `stranid` int(11) NOT NULL DEFAULT '0',
  `red` int(11) NOT NULL DEFAULT '0',
  `file` varchar(255) DEFAULT NULL,
  `opis` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`galerijaid`),
  KEY `stranid` (`stranid`),
  KEY `atributid` (`atributid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `galerija`
--
-----

--
-- Table structure for table `podatek`
--

CREATE TABLE IF NOT EXISTS `podatek` (
  `stranid` int(11) NOT NULL DEFAULT '0',
  `atributid` int(11) NOT NULL DEFAULT '0',
  `ivrednost` int(11) NOT NULL DEFAULT '0',
  `tvrednost` varchar(255) DEFAULT NULL,
  `vvrednost` blob,
  `rvrednost` float NOT NULL DEFAULT '0',
  KEY `atributid` (`atributid`),
  KEY `stranid` (`stranid`)
)

```

```

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `podatek`
--

-----
-- Table structure for table `polje`
--
CREATE TABLE IF NOT EXISTS `polje` (
  `poljeid` int(11) NOT NULL AUTO_INCREMENT,
  `size` int(11) NOT NULL DEFAULT '0',
  `privzeto` int(11) NOT NULL DEFAULT '0',
  `vprasanje` varchar(255) DEFAULT NULL,
  `poljetip` int(11) NOT NULL DEFAULT '0',
  `obvezen` int(11) NOT NULL DEFAULT '0',
  `red` int(11) NOT NULL DEFAULT '0',
  `stranid` int(11) NOT NULL DEFAULT '0',
  `atributid` int(11) NOT NULL DEFAULT '0',
  `special` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`poljeid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `polje`
--

-----

--
-- Table structure for table `pripadnost`
CREATE TABLE IF NOT EXISTS `pripadnost` (
  `atributid` int(11) NOT NULL DEFAULT '0',
  `stranid` int(11) NOT NULL DEFAULT '0',
  `targetid` int(11) NOT NULL DEFAULT '0'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `pripadnost`
--

-----

--
-- Table structure for table `stran`
--
CREATE TABLE IF NOT EXISTS `stran` (
  `stranid` int(11) NOT NULL AUTO_INCREMENT,
  `parent` int(11) NOT NULL DEFAULT '0',
  `uporabnikid` int(11) NOT NULL DEFAULT '0',
  `tipid` int(11) NOT NULL DEFAULT '0',
  `naslov` varchar(255) DEFAULT NULL,
  `modified` int(11) NOT NULL DEFAULT '0',
  `red` int(11) NOT NULL DEFAULT '0',
  `deleted` int(11) NOT NULL DEFAULT '0',
  `skrij` int(11) NOT NULL DEFAULT '0',
  `zaklenjen` int(11) NOT NULL DEFAULT '0',
  `sellable` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`stranid`),
  KEY `parent` (`parent`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `stran`
--

-----

--
-- Table structure for table `tip`
--
CREATE TABLE IF NOT EXISTS `tip` (
  `tipid` int(11) NOT NULL AUTO_INCREMENT,
  `naziv` varchar(255) DEFAULT NULL,
  `deleted` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`tipid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `tip`
--

```