



PHP Web Development

ZAVRŠNI RAD

Mentor:
Prof. Vladimir Marić

Kandidat:
Ivan Jojić G2

Beograd, 2016

Sadržaj:

1.Uputstvo za korisnike ...3	
1.1.Logo i navigacija ...3	
1.2.Front strane ...4	
1.3.Sidebar ...4	
1.4.Početna ...5	
1.5.O Nama ...5	
1.6.Usluge ...6	
1.7.Reference ...6	
1.8.Cenovnik ...7	
1.9.Kontakt ...7	
2.Uputstvo za programere ...9	
2.1.HTML i CSS ...9	
2.2.PHP i MySQL ...11	
2.2.1.config.php ...11	
2.2.2.index.php ...12	
2.2.3.Connection.class.php ...12	
2.2.4.Base.class.php ...13	
2.2.5.Article.class.php ...14	
2.2.6.Poster.class.php ...16	
2.2.7.Price.class.php ...18	
2.2.8.Page.class.php ...20	
2.2.9. .htaccess ...22	
2.2.10.send.php ...22	
2.3.JavaScript ...23	

1.Uputstvo za korisnike

Sajt IVI-SISTEM predstavlja građevinsko preduzeće, sa svim informacijama za potrebe budućih klijenata i poslovnih saradnika.

Strane:

- 1.Početna
- 2.O Nama
- 3.Usluge
- 4.Reference
- 5.Cenovnik
- 6.Kontakt

1.1.Logo i navigacija

U okviru svake strane u gornjem delu nalazi se logo sa navigacijom. Navigacija se sastoji od tastera koji su u svojoj pozadini linkovi koji vode ka drugim stranama sajta. Logo koji je jedna vrsta slike u svojoj pozadini je takođe link koji vodi, u ovom slučaju, na početnu stranu.

Navigacija i logo su postavljeni u horizontalnom položaju, i klikom na neki od tastera idete na traženu stranu sajta.

Navigacija je indentična na svim stranama.



1.2.Front strane

Front strane predstavljaju tri slike. Te slike su neke od referenci odnosno objekata na kojima smo radili. Slike su predstavljene kao slajd, odnosno na svakih pet sekundi druga slika zamenjuje prvu, treća zamenjuje drugu a prva zamenjuje treću itd.. Svo vreme prisrupa korisnika na sajtu, slajd radi neprekidno.

Kao i navigacija front je indentičan na svim stranama.



1.3.Sidebar

U Sidebar-u nalaze se reklame odnosno zakupljena mesta na određeno vreme od određenog proizvođača materijala. Neke od reklama su zakupljene novcem, a pojedine su dodeljene na osnovu pogodnosti kojima proizvođač daje popust pri nabavci materijala nama kao građevinskom preduzeću.

Sidebar sa reklamama je postavljen vertikalno, i on predstavlja niz slika koji su u pozadini linkovi koji vode ka sajtu zakupca reklamnog prostora.

Kao kod navigacije i fronta, i sidebar se nalazi na svakoj strani sajta.



1.4. Početna

Na početnoj strani nalaze se četiri članka.

U prvo članku se nalaze osnovna obaveštenja o radovima koje izvodimo kao i o tome šta možete očekivati od poslovne saradnje sa nama.

Naredna tri članka predstavljaju ono o čemu se susreće svaki potencionalni klijent pri odluci da renovira poslovni prostor, stambeni prostor ili dvorište.

Adaptacija poslovnog prostora



Kancelarija je zapravo vaša druga kuća. To je mesto gde se borite za nove poslovne prilike, kreirate i ostvarujete svoje profesionalne ambicije. Radni prostor takođe ima veliki udeo u identitetu firme, funkcionisanju tima i u tome kako će ga doživeti vaši klijenti. Jedan od uslova dobre prodaje je i atraktivan poslovni prostor, bilo da se radi o prodajnoj ili uslužnoj delatnosti. Estetika, funkcionalnost i pozitivna energija prostora sigurno će doprineti da se vaši klijenti osećaju prijatno i opušteno u Vašem prostoru i da svoje poverenje i novac poklone upravo Vama. Kvalitetna adaptacija poslovnog prostora (kancelarije, lokala, firme) podrazumeva da počnete sa dobrim projektom adaptacije enterijera i eksterijera i arhitektom od poverenja, koji će za Vas voditi sve poslove adaptacije.



1.5. O Nama

Na strani 'o nama' nalaze se dva članka koji govore o istoriji preduzeća, našim smernicama, vrednostima kojima stremimo i budućim projektima koji su pred nama.

IVI-SISTEM



IVI-SISTEM posluje već duži niz godina, usavršavajući se u skladu sa savremenim načinom življenja i potrebom da se kvalitetno, funkcionalno i estetski odgovori zahtevima kupaca. Osnivači i vlasnici firme su Vida Stanisic i Ivan Jojic.

Firma IVI-SISTEM je osnovana 01.09.2010 godine u Beogradu i od tada zahvaljujući kvalitetu i upornom angažovanju stručnjaka u svim oblastima adaptacija i renoviranja poslovnih, stambenih prostora postiže izuzetne rezultate o kome govori sekocija referencija gde je samo mali broj veoma velikih firmi sa kojima saradjujemo.

Početak rada ove firme bio je sa malim brojem ljudi, iz godine u godinu ova firma je znatno napredovala tako da je izrasla u moćnu radnu organizaciju sposobnu da samostalno kvalitetno i na vreme obavi i najsloženije građevinske zahvate. Zapošljavaju se vrhunski

majstori, koji svojom stručnošću odrade svaki posao na najbolji mogući način. Sva prethodno stečena znanja i iskustva kroz rad na značajnim objektima i poslovima uneta su kao značajan kapital i tokom godina prenošena na mlade kadrove. Proteklih godina prošli smo kroz mnoge poteškoće poslujući u teškoj ekonomskoj situaciji u našoj zemlji, da bi smo iz toga izašli još jači, i u povoljnijim ekonomskim uslovima u potpunosti pokazali svoje potencijale i na tržištu zauzeli mesto koje nam pripada. Visoka stručnost operativne, zajedno sa menadžmentom preduzeća, omogućili su kvalitetno izvođenje radova u svim uslovima, a prema standardima i zahtevima investitora. Zahvaljujući tome IVI-SISTEM postaje prepoznatljiv brend, a stečeni renome bazira na bogatoj referencijalisti izvedenih objekata.

Verujemo u svoje ljude, ni u ovim teškim vremenima nećemo posustati u poslovima koji su pred nama.



1.6. Usluge

Na strani 'Usluge' nalazi se sedam članaka o osnovnim uslugama koje pruža preduzeće. Svaka vrsta radova je ukratko opisana, sa dodatnim pojašnjenjem kako možemo da Vam pomognemo pri adaptaciji.

Prostorno planiranje



Prilikom adaptacije imamo tačno određen raspored obavljanja poslova. Tako radeći ostvarujemo najbolje rezultate. Prva faza adaptacije je svakako prostorno planiranje. Poštujući Vaše želje svaki kutak Vašeg doma ćemo učiniti prijatnim i funkcionalnim. Izradimo prečizan plan prostora. Ukoliko je potrebno, možemo ukloniti suvišne zidove ili zidati nove. Radimo i zamenu elektroinstalacija, vodovodnih instalacija, stolarije, podnih obloga... Trudimo se da Vaš dom bude mesto u kom se najbolje osećate. Ako budete angažovali nas, budete sigurni da ćemo isplanirati svaki pa i najmanji detalj u Vašem domu. Nećete se više saplitali o kablove, imaćete dovoljno utičnica na svim mestima u domu, u zavisnosti od namene prostorije predložićemo Vam boju u koju ćete je okrećiti, uradićemo zajedno plan rasporeda sanitarija, a po Vašoj želji možemo Vam pomoći i u izboru nameštaja.

Šuva gradnja



Rigips pregradni zidovi imaju iste, ako ne i bolje zvučne, toplotne i protivpožarne osobine kao i zidovi od o cigle ili blokova. Prednost Rigips zidova je u tome što se lako postavljaju, održavaju, lakši su i lako se demontiraju. Šupljina zida se može iskoristiti za sprovođenje svih instalacija. Gipsanim pločama se rade i spuštani plafoni. U prostor između ploča i plafona lako se mogu namontirati sistemi provetravanja i klimatizacije, a dodavanjem rasvete se mogu zadovoljiti visoki estetski zahtevi.



1.7. Reference

Na strani 'reference' nalazi se pet članaka. U njima su predstavljeni neki od projekata na kojima smo radili proteklih godina. Izdvojeni su samo oni najveći.

NPS 16 - Ruska Federacija - Jakutska oblast



NPS 16 je bazna stanica, u istočnom Sibiru, za transport nafte u okviru Transneft korporacije. Transneft je Ruski operater magistralnih naftovoda. On je najveća svetska naftna kompanija, poseduje 48,7 hiljada kilometara magistralnih cevovoda, više od 500 pumpnih stanica.

U sklopu NPS 16 nalazi se postrojenje (pumpa) za ubrzani transport nafte, rezervoari za skladištenje nafte, toplana za grejanje objekata, trafo stanica sa dalekovodima za prenos električne energije, garaža za mehanizaciju, objekat vodovoda, kanizacioni sistem, hoteli za smeštaj zaposlenih kao i mnogi drugi prateći objekti.



1.8.Cenovnik

Na strani 'cenovnik' nalazi se jedan članak i tabela. Tabela se sastoji od četiri kolone. U prvoj koloni nalazi se redni broj pozicije. Druga kolona je vrsta radova. A u trećoj i četvrtoj koloni su mera i cena po jedinici mere.

Cenovnik radova



U priloženom cenovniku se nalazi samo deo asortimana raspoloživih usluga sa kratkim opisom. Cene koje su date mogu se smatrati samo informativnim, zbog velike oscilacije zahteva i potreba klijenata, nivoa detaljnosti i kompleksnosti objekta. Konačnu ponudu možete dobiti samo nakon izlaska našeg stručnog tima na teren. Ukoliko imate ušestale potrebe za izvođenjem molerskih, gipsarskih ili dekorativnih radova, obaveštavamo Vas da za stalne klijente imamo poseban popust na sve usluge. Pozovite nas ili nam pošaljite na e-mail Vaš zahtev za izvođenje potrebnih radova i mi ćemo Vam u kratkom roku dati okvirnu ponudu koja sadrži cenu rada i potrebnog materijala, kao i rokove potrebnih za završetak radova. Upoznavanje sa projektom (ukoliko postoji), definisanje svih aspekata vezano za radove koje želite da se izvedu. Početak izvođenja radova, ušestakost narednih konsultacija najviše zavise od kompleksnosti potrebnih radova. Nakon završetka radova vršimo pripremu prostora.

BR	VRSTA RADOVA	MERA	CENA
1	Lepljenje ploča na zidove - suvo materisanje	m2	7 eur
2	Termo i zvučno izolaciono oblaganje zidova sa metalnom potkonstrukcijom	m2	13 eur
3	Oblaganje plafona sa potkonstrukcijom	m2	12 eur
4	Oblaganje potkrovlja sa potkonstrukcijom	m2	16 eur

KNAUF

Rigips
SAINT-GOBAIN

KERAMIKA
KANJIŽA
PLUS

1.9.Kontakt

Na strani 'kontakt' nalazi se jedan članak, adresa sa mogućnošću da se klikom na e-mail direktno pokrene podrazumevani desktop program za slanje i primanje e-mail pošte.

Kontakt

Kako bi nas kontaktirali popunite formular ispod. Mi ćemo odgovoriti u najkraćem mogućem roku. Molimo Vas da popunite pravilno sve informacije, tako da možemo da vas kontaktiramo.

IMI-SISTEM:

Sremačka 3/a, Vršin, Beograd
Tel: 064/183-50-36
Email: ivan.jojic81@gmail.com
Net: www.imi-sistem.rs

KNAUF

Rigips
SAINT-GOBAIN

Na sredini 'kontakt' strane nalazi se formular sa koga se može poslati poruka preduzeću. Poruka može sadržati neko pitanje u vezi radova ili rešiti nedoumice u vezi predhodnih delova sajta.

Polje za e-mail je obavezno, u njemu se mora uneti ispravna e-mail adresa. U slučaju greške kod unosa e-mail adrese pojavljuje se obaveštenje o grešci i taster za ponavljanje celog procesa slanja poruke.

Kontakt

Ime:

E-mail:

Pošalji poruku:

Pošalji

MI-SISTEM:

Sremačka 3/a, Vrčin, Beograd
Tel: 064/183-50-36
Email: Ivan.Jojlo81@gmail.com
Net: www.MI-sistem.rs



Na kraju 'kontakt' strane nalazi se Google mapa, na kojoj je označena tačna pozicija preduzeća. Ova mapa znatno olakšava klijentima i poslovnim partnerima orijentaciju pri pronalaženju našeg predstavništva.

Scroll-ovanjem točkića na mišu, mapa se može uvećavati ili smanjivati po potrebi.

Lokacija



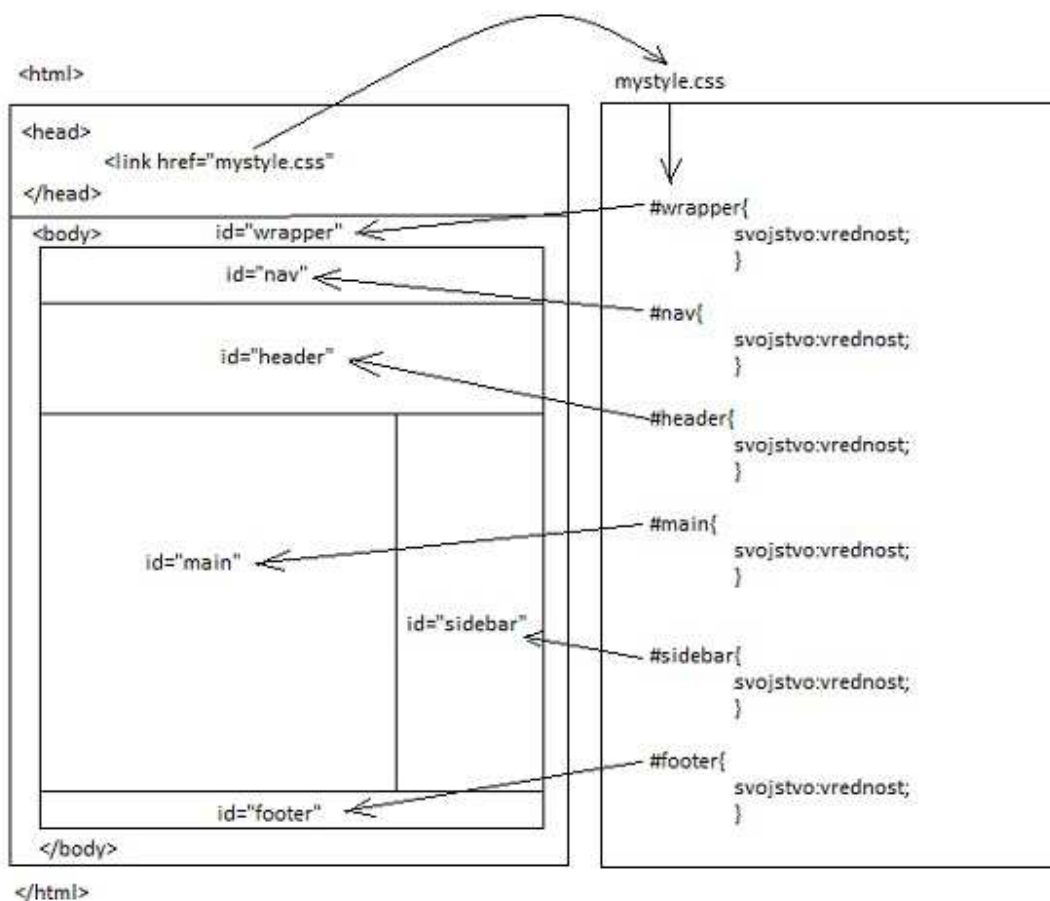
2.Uputstvo za programere

U uputstvu za programere nalazi se opis funkcionisanja sajta pomoću programskih jezika HTML-a, CSS-a, PHP-a, JavaScript-a kao i baze podataka MySQL-a.

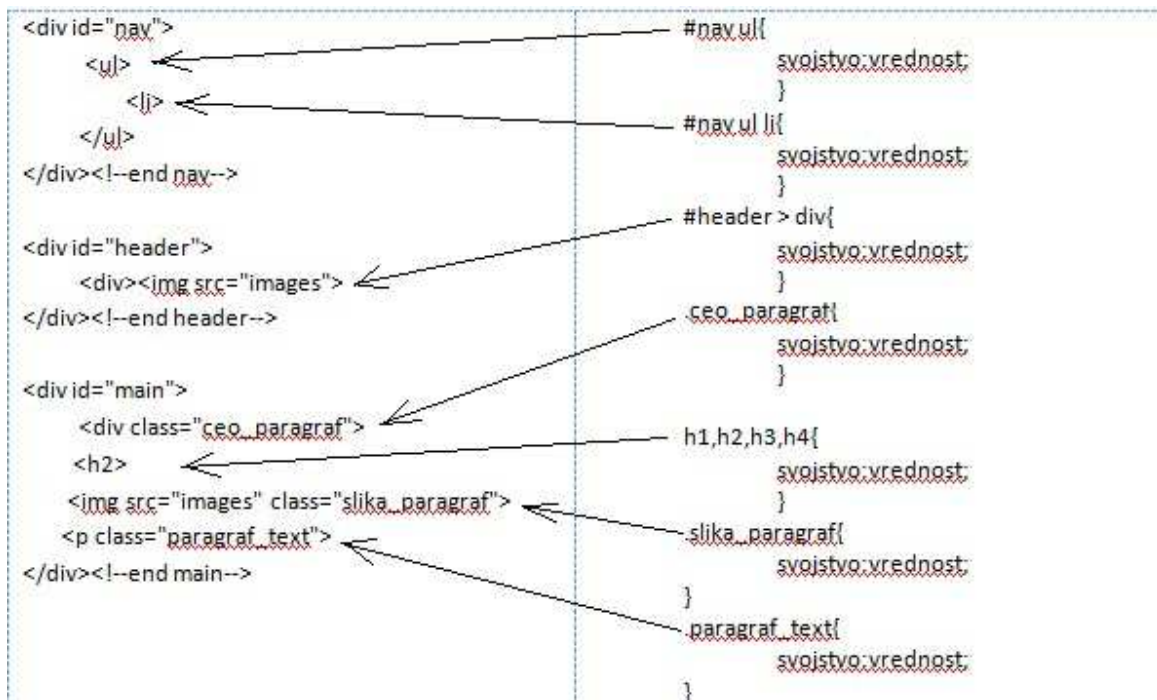
2.1.HTML i CSS

HTML (Hyper Text Markup Language) čini kostur i osnovu strane, dok CSS(Cascading Style Sheets) daje elementima HTML-a poziciju i boju, odnosno način na koji će se oni prikazati na strani.

U ovom sajtu se koristi eksterni CSS fajl koji se preko linka na HTML strani u zaglavlju stranice između <head> taga povezuje sa određenim selektorima.



Unutar <div> tagova koji čine osnovu strane nalaze se elementi koji su takođe stilizovani preko svoji selektora koji se nalaze u CSS fajlu.



Na kraju CSS fajla nalazi se '**Responsive dizajn**' koji prilagođava prikaz, delova strane, veličini browser-a. Ukoliko korisnik pređe sa računara na tablet ili mobilni telefon, sajt se prilagođava i vrši se preraspodela elemenata radi potrebne funkcionalnosti.

Responsive dizajn se sastoji od tri **media upita**:

- za prikaz između 1024 i 1260 piksela

```
@media screen and (min-width:1024px) and (max-width:1260px) {
    svojstvo:vrednost;
}
```

- za prikaz između 768 i 1023 piksela

```
@media screen and (min-width:768px) and (max-width:1023px) {
    svojstvo:vrednost;
}
```

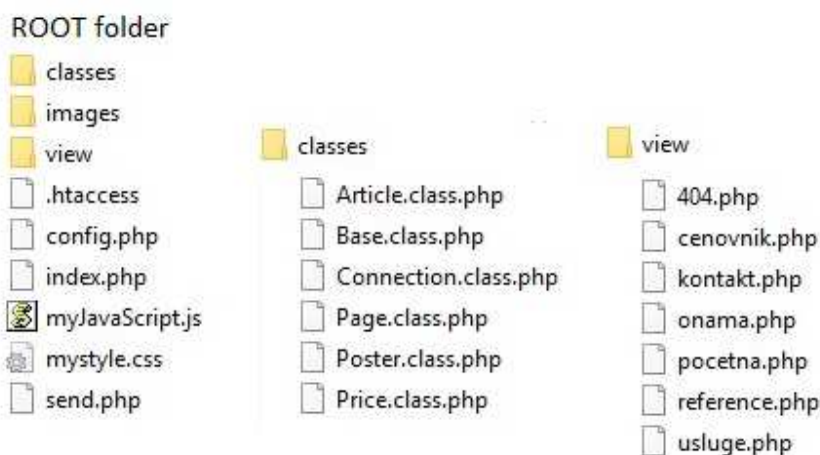
- za prikaz manji od 767 piksela

```
@media screen and (max-width:767px) {
    svojstvo:vrednost;
}
```

2.2.PHP i MySQL

Veći deo sadržaja sajta prikazuje se i dobavlja pomoću PHP-a. Iz MySQL baze podataka PHP dobavlja sadržaj i prikazuje ga na stranama sajta.

Izgled root foldera sajta:



2.2.1.config.php

Pošto su podaci za konekciju sa bazom nepromenjivi, u config.php fajlu nalaze se definisane konstante za putanju do root foldera, lokalnu IP adresu, korisničko ime i šifra za bazu podataka kao i sam naziv baze podataka.

Na kraju config.php fajla nalazi se funkcija `__autoload()`, koja se koristi za automatsko učitavanje fajla, odnosno za učitavanje onoga što je potrebno iz tog fajla. Ime klase koja se poziva i ime fajla u kome se nalazi klasa moraju biti indentični.

Kao parametar za funkciju `autoload()` koristi se promenljiva `$cname`, koja pomoću `'require_once'` učitava klase iz foldera `'classes'`. `'require_once'` ubacuje jedan sadržaj php fajla u drugi php fajl, s tim što ako je fajl već učitao ignorisaće sve ostale zahteve za učitavanje istog fajla.

```
config.php
<?php
define("APP_DIR","C:/wamp64/www/ivi-sistem/");
define("DBHOST","localhost");
define("DBUSER","root");
define("DBPASS","");
define("DBNAME","ivi-sistem");
function __autoload($cname){
    require_once "classes/{$cname}.class.php";
}
```

2.2.2.index.php

Nakon učitavanja fajla config.php u fajl index.php pomoću 'require', ovaj fajl postaje ruter u kome se odigrava veći deo korisničkog procesa.

U daljem tekstu prikazaće se detaljno funkcionisanje fajla index.php.

A screenshot of a code editor showing the content of index.php. The code is in PHP and includes a require statement for config.php, an if statement checking for the 'strana' GET parameter, and an include statement for view/pocetna.php.

```
index.php x
<?php
require "config.php";

if(isset($_GET['strana'])) {
    $p = Page::get($_GET['strana']);
    Page::loadView($p->name);
}else{
    include "view/pocetna.php";
}

?>
```

2.2.3.Connection.class.php

Ovaj fajl sadrži klasu Connection koja je 'singleton pattern'. Fajl Connection.class.php će preko fajla config.php i funkcije __autoload() učitava u fajl index.php. Klasa se poziva samo jednom, odnosno ako konekcija već postoji vraća tu istu konekciju.

U klasi Connection se nalazi statička metoda getConnection() koja pomoću PDO-a pristupa podacima iz baze podataka. Kao parametri za PDO koriste se definisane konstante iz fajla config.php.

A screenshot of a code editor showing the content of Connection.class.php. The code defines a Connection class with a private static \$conn variable and a public static getConnection() method that uses PDO to connect to a database.

```
Connection.class.php x
<?php
class Connection{
    private static $conn = null;
    public static function getConnection(){
        if (!self::$conn) {
            self::$conn = new PDO("mysql:host=".DBHOST."
                                ;dbname=".DBNAME.";charset=utf8",DBUSER,
                                DBPASS);
        }
        return self::$conn;
    }
}
```

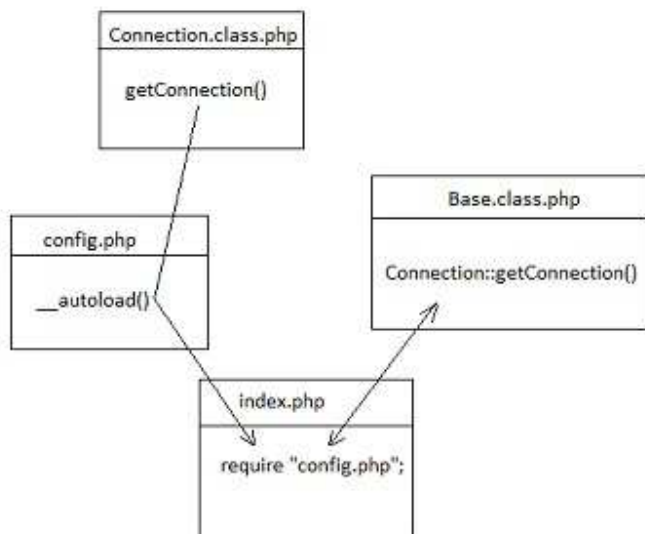
2.2.4.Base.class.php

U ovo fajlu nalazi se **abstraktna klasa Base** sa statičnim metodama **get()** i **remove()**. Ova klasa može da se nasleđuje ali ne može da se instancira.

Metode **get()** i **remove()** služe za dobavljanje odnosno za uklanjanje podataka iz baze.

Obe sadrže kao parametar promenjivu **\$id** koja u MySQL upitu zamenjuje kolonu **id**(primary key) iz tabele u bazi.

Klase koje nasleđuju klasu Base, to vrše tako što ta klasa preko fajla **config.php** i funkcije **autoload()** dobavlja konekciju iz metode **getConnection()** iz klase **Connection** u fajlu **index.php**.



```
Base.class.php
<?php
abstract class Base{
    static function get($id){
        $className = get_called_class();
        $conn = Connection::getConnection();
        $tableName = static::$tableName;
        $res = $conn->query("SELECT * FROM {$tableName} WHERE id={$id}");
        $res->setFetchMode(PDO::FETCH_CLASS,$className);
        $rw= $res->fetch();
        return $rw;
    }
    static function remove($id){
        $conn = Connection::getConnection();
        $tableName = static::$tableName;
        $conn->query("DELETE FROM {$tableName} WHERE id='{$id}'");
    }
}
```


U metodama `get()` i `remove()` nalazi se:

`$className` -promenljiva koja pomoću metode `get_called_class()` dobavlja ime klase i unosi u `setFetchMode()`

`$tableName` -promenljiva koja će kod kasnije nasleđenih klasa u MySQL upitu zamenjivati ime tabele iz baze. Ona predstavlja statičko polje klase u kojoj se nalazi

`setFetchMode()` - menja podrazumevani PDO

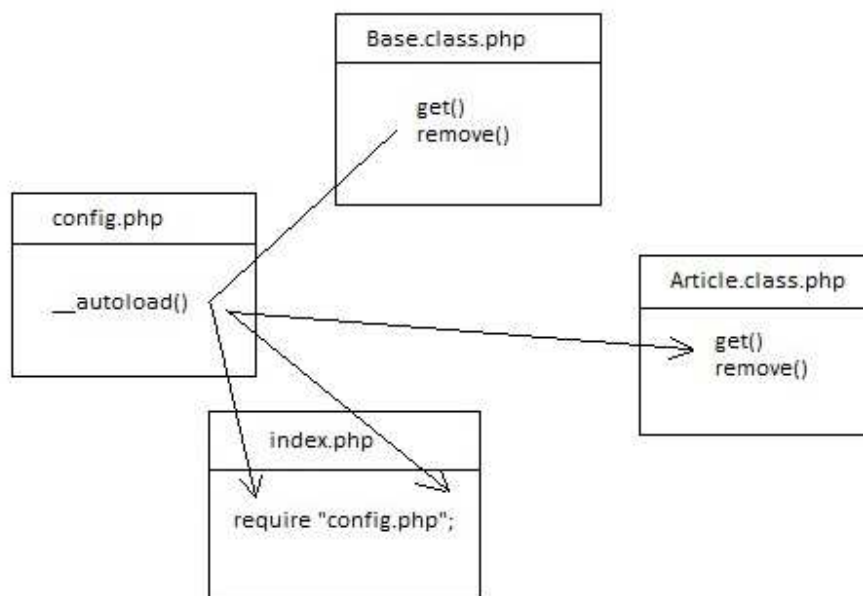
`fetch()` -prikazuje red iz tabele kao klasu

```
query("SELECT * FROM {$tableName} WHERE id={$id}");
```

```
setFetchMode(PDO::FETCH_CLASS,$className);
```

2.2.5.Article.class.php

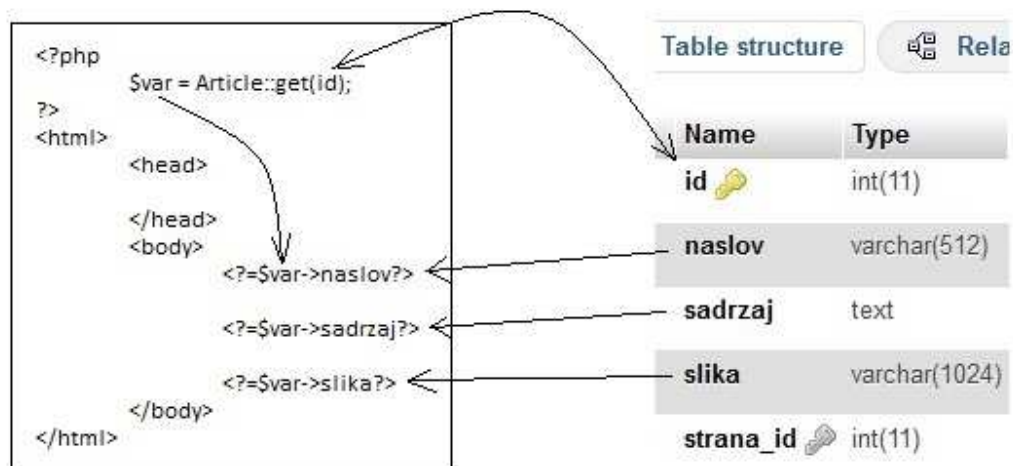
U ovo fajlu nalazi se klasa `Article` koja nasleđuje klasu `Base`, odnosno statičke metode **`get()`** i **`remove()`**, za dobavljanje i brisanje redova tabele iz baze podataka. Nasleđivanje klase `Base` se obavlja preko fajlova `index.php` i `config.php`, metodom `autoload()`.



```
Article.class.php
<?php
class Article extends Base{
    static $tableName = "clanci";
    function insert(){
        $conn = Connection::getConnection();
        $conn->query("INSERT INTO clanci VALUES(null,'{$this->naslov}','
            {$this->sadrzaj}','{$this->slika}','{$this->strana_id}')");
    }
}
```

Klasa Article isključivo dobavlja podatke iz tabele 'clanci' iz baze podataka. U klasi postoji statičko polje **\$tableName**, koje nosi vrednost 'clanci', a koje u upitu za MySQL menja naziv tabele.

Istanciranjem klase Article, odnosno pravljenjem novog objekta klase Article, sa statičkom metodom **get()** i dodavanjem parametra kao **id** reda iz tabele 'clanci', na samom početku koda na strani, dobija se mogućnost prikazivanja podataka iz tabele, unutar HTML koda.



Brisanje reda iz tabele 'clanci' obavlja se instanciranjem klase Article sa statičkom metodom **remove()**, koja sadrži parametar koji predstavlja **id** reda iz tabele.

Article::remove(id);

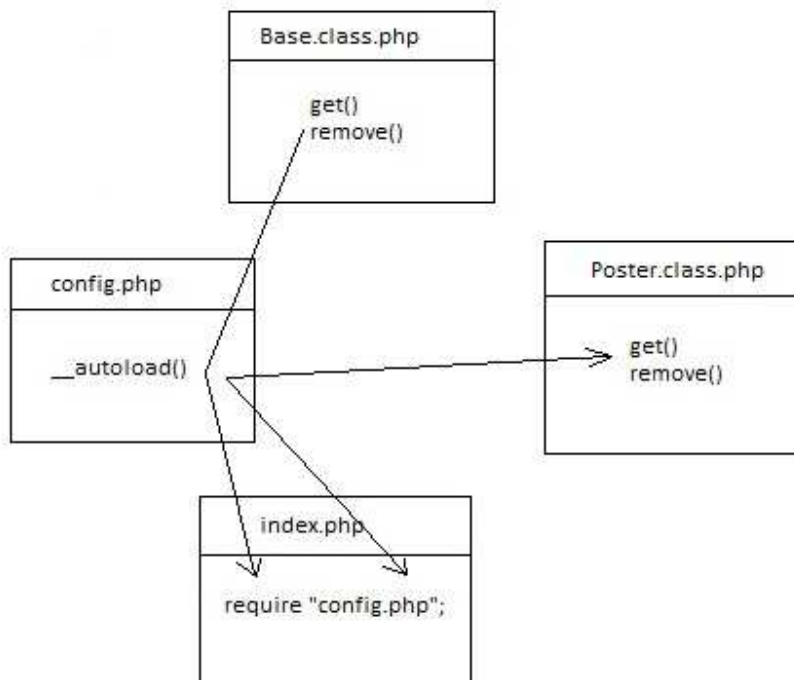
Osim ove dve nasleđene metode, klasa Article sadrži i svoju metodu **insert()**, koja služi za unošenje podataka u bazu.

Potrebno je napraviti novi objekat klase Article, i uneti vrednosti za određenu kolonu u tabeli 'clanci'. Za unošenje novih podataka najpogodniji je index.php jer je on ruter fajl.



2.2.6. Poster.class.php

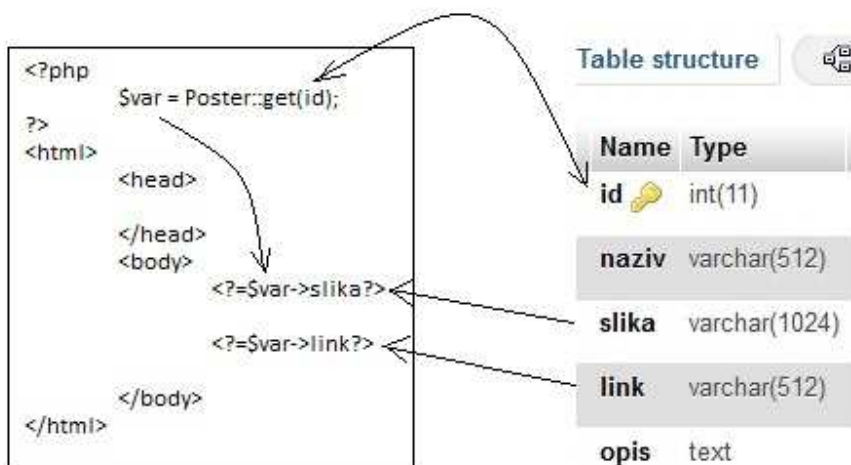
U ovoj fajlu nalazi se klasa Poster koja nasleđuje klasu Base, odnosno statičke metode **get()** i **remove()**, za dobavljanje i brisanje redova tabele iz baze podataka. Nasleđivanje klase Base se obavlja preko fajlova index.php i config.php, metodom `autoload()`.



```
Poster.class.php
<?php
class Poster extends Base{
    static $tableName = "reklame";
    function insert(){
        $conn = Connection::getConnection();
        $conn->query("INSERT INTO reklame VALUES(null, '{ $this->naziv}',
            '{ $this->slika}', '{ $this->link}', '{ $this->opis}')");
    }
}
```

Klasa Poster isključivo dobavlja podatke iz tabele **'reklame'** iz baze podataka. U klasi postoji statičko polje **\$tableName**, koje nosi vrednost **'reklame'**, a koje u upitu za MySQL menja naziv tabele.

Istanciranjem klase Poster, odnosno pravljenjem novog objekta klase Poster, sa statičkom metodom **get()** i dodavanjem parametra kao **id** reda iz tabele **'reklame'**, na samom početku koda na strani, dobija se mogućnost prikazivanja podataka iz tabele, unutar HTML koda.



Brisanje reda iz tabele 'reklame' obavlja se instanciranjem klase Poster sa statičkom metodom **remove()**, koja sadrži parametar koji predstavlja **id** reda iz tabele.

Poster::remove(id);

Osim ove dve nasleđene metode, klasa Poster sadrži i svoju metodu **insert()**, koja služi za unošenje podataka u bazu.

Potrebno je napraviti novi objekat klase Poster, i uneti vrednosti za određenu kolonu u tabeli 'reklame'. Za unošenje novih podataka najpogodniji je index.php jer je on ruter fajl.



Pošto se sve reklame u sidebar-u pojavljuju na svakoj strani sajta, da bi se smanjio kod, odnosno da bi se izbeglo nepotrebno instanciranje objekta klase Poster, napravljen je niz od kolona iz tabele 'reklame' koji prikazuje 'slike' i 'linkove'.

Pomoću **foreach** petlje i **fetchAll()** koji vraća niz sa svim rezultatima redova iz tabele 'reklame' u nastavku je prikazan u okviru shorthand koda dobavljanje pdataka iz asocijativnog niza **\$var['key']**.

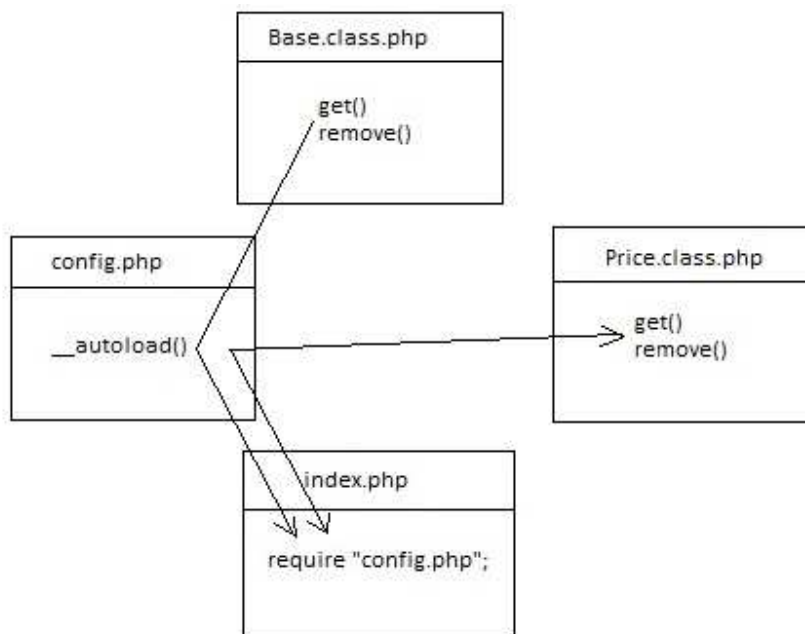
```

<div id="sidebar">
    <?php
        $conn = Connection::getConnection();
        $res = $conn->query("SELECT * FROM reklame");
        foreach ($res->fetchAll() as $rw) { ?>
            <a href="<?=$rw['link']?>" target="_blank">
                </a>
        } ?>
    </div><!--end sidebar-->

```

2.2.7.Price.class.php

U ovo fajlu nalazi se klasa Price koja nasleđuje klasu Base, odnosno statičke metode **get()** i **remove()**, za dobavljanje i brisanje redova tabele iz baze podataka. Nasleđivanje klase Base se obavlja preko fajlova index.php i config.php, metodom **autoload()**.



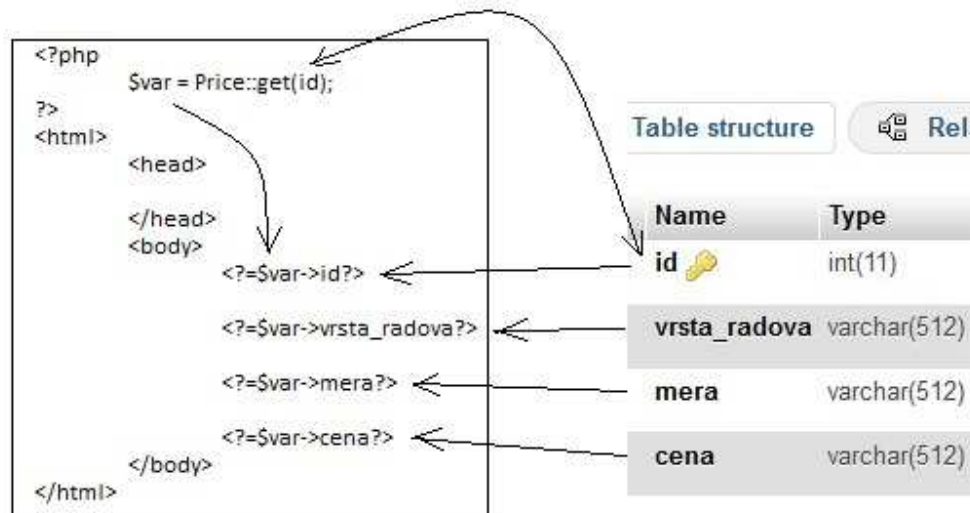
```

Price.class.php
<?php
class Price extends Base{
    static $tableName = "cenovnik";
    function insert(){
        $conn = Connection::getConnection();
        $conn->query("INSERT INTO cenovnik VALUES(null, '{$this->vrsta_radova}',
            '{$this->mera}', '{$this->cena}')");
    }
}

```


Klasa Price isključivo dobavlja podatke iz tabele 'cenovnik' iz baze podataka. U klasi postoji statičko polje **\$tableName**, koje nosi vrednost 'cenovnik', a koje u upitu za MySQL menja naziv tabele.

Istanciranjem klase Price, odnosno pravljenjem novog objekta klase Price, sa statičkom metodom **get()** i dodavanjem parametra kao **id** reda iz tabele 'cenovnik', na samom početku koda na strani, dobija se mogućnost prikazivanja podataka iz tabele, unutar HTML koda.

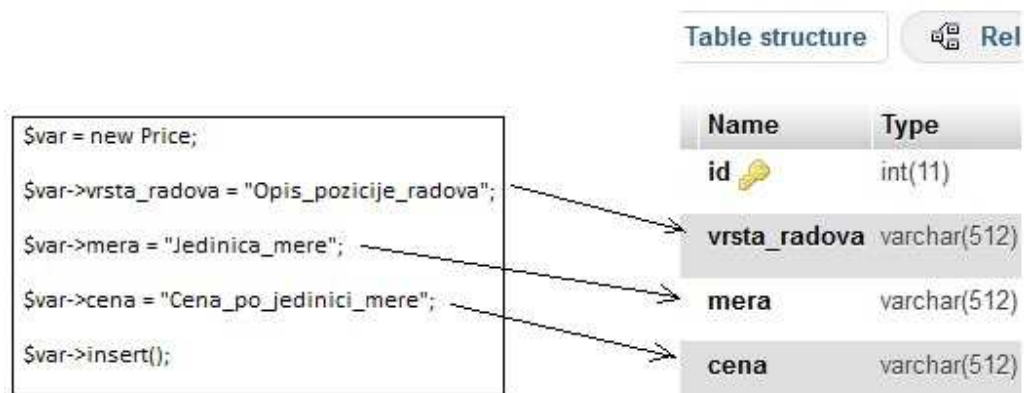


Brisanje reda iz tabele 'cenovnik' obavlja se instanciranjem klase Price sa statičkom metodom **remove()**, koja sadrži parametar koji predstavlja **id** reda iz tabele.

Price::remove(id);

Osim ove dve nasleđene metode, klasa Price sadrži i svoju metodu **insert()**, koja služi za unošenje podataka u bazu.

Potrebno je napraviti novi objekat klase Price, i uneti vrednosti za određenu kolonu u tabeli 'cenovnik'. Za unošenje novih podataka najpogodniji je index.php jer je on ruter fajl.



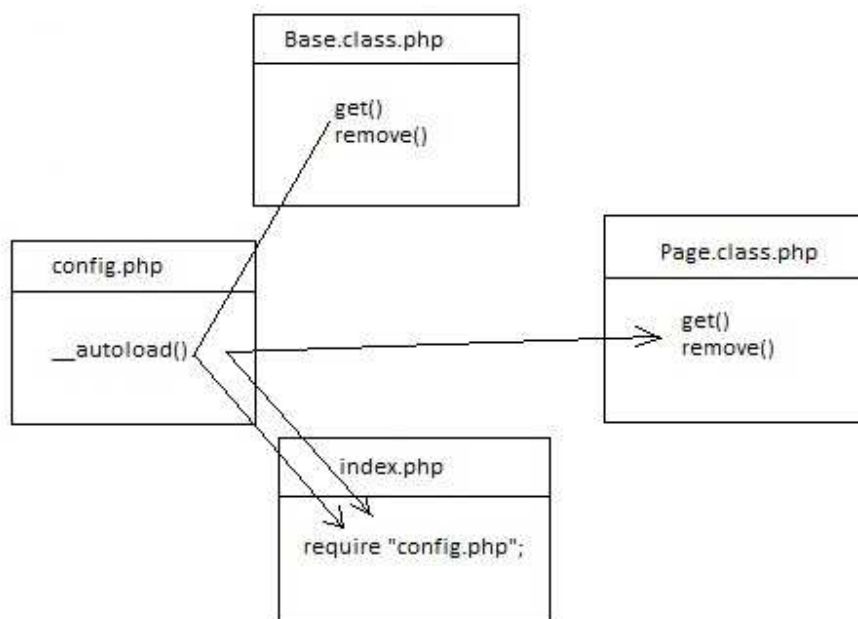
Pošto se ceo cenovnik prikazuje na strani cenovnik.php, a sadrži veliki broj podataka, da bi se smanjio kod, odnosno da bi se izbeglo nepotrebno instanciranje objekta klase Price, napravljen je niz od kolona iz tabele 'cenovnik' koji prikazuje 'broj reda', 'vrstu radova', 'meru' i 'cenu' u okviru tabele.

Pomoću **foreach** petlje i **fetchAll()** koji vraća niz sa svim rezultatima redova iz tabele 'cenovnik' u nastavku je prikazan u okviru shorthand koda dobavljanje podataka iz asocijativnog niza **\$var['key']**.

```
<table>
  <tr><th>BR</th><th>VRSTA  RADOVA</th><th>MERA</th><th>CENA</th></tr>
  <?php
  $conn = Connection::getConnection();
  $res = $conn->query("SELECT * FROM cenovnik");
  foreach ($res->fetchAll() as $rw) { ?>
    <tr>
      <td><?=$rw['id']?></td>
      <td><?=$rw['vrsta_radova']?></td>
      <td><?=$rw['mera']?></td>
      <td><?=$rw['cena']?></td>
    </tr>
  <?php } ?>
</table>
```

2.2.8.Page.class.php

U ovo fajlu nalazi se klasa Page koja nasleđuje klasu Base, odnosno statičke metode **get()** i **remove()**, za dobavljanje i brisanje redova tabele iz baze podataka. Nasleđivanje klase Base se obavlja preko fajlova index.php i config.php, metodom autoload().



Osim dve nasleđene metode, klasa Page ima i metodu **loadView()**, koja služi za učitavanje fajlova iz view foldera. Kao parametar koristi promenjivu **\$viewName** koja u telu metode pri učitavanju fajla zamenjuje naziv strane.

```
Page.class.php
<?php
class Page extends Base{
    static $tableName = "strane";
    static function loadView($viewName){
        include "view/{$viewName}.php";
    }
}
```

Klasa Page isključivo dobavlja podatke iz tabele 'strane' iz baze podataka. Istanciranjem klase Page, odnosno pravljenjem novog objekta klase Page, sa statičkom metodom **get()** na strani index.php dobavlja se red iz tabele 'strane' preko id.

Promenjiva \$id koja predstavlja red iz tabele u bazi, zamenjena je parametrizacijom **\$_GET['strana']**, tako da unošenjem putanje u address bar **/?strana=id_iz_tabele** dobijamo red tabele u vidu novog objekta. Vrednost polja novog objekta 'name' koje predstavlja kolonu iz tabele ubacuje se kao parametar u metodu loadView iz klase Page. Rezultat ove kontrole toka biće da je strana pocetna.php -> ?strana=1, onama.php -> ?strana=2, usluge.php -> ?strana=3 itd... Kasnije će u fajlu .htaccess ove putanje biti skrivene.

U slučaju da ne postoji parametar učitace se fajl view/pocetna.php.

```
index.php
<?php
require "config.php";

if(isset($_GET['strana'])) {
    $page = Page::get($_GET['strana']);
    Page::loadView($page->name);
}else{
    include "view/pocetna.php";
}
```

Table structure

Name	Type
id	int(11)
name	varchar(512)
link	varchar(512)
opis	text

2.2.9. .htaccess

Ovo je fajl koji menja Apache server, odnosno prilagođava ga korisniku.

Pomoću komande RewriteRule skrivena je prava putanja do stranica dobijena parametrizacijom, tako da će se sada putanja do fajla pocetna.php umesto ?strana=1 videti POCETNA itd...

U slučaju da korisnik preko address bar-a zatraži nepostojeću stranicu, umesto klasičnog Apache obaveštenja 'ErrorDocument 404' prikazaće se sadržaj fajla 404.php, sa obaveštenjem da stranica nije pronađena.

```
.htaccess
RewriteEngine On
RewriteRule POCETNA ?strana=1
RewriteRule O_NAMA ?strana=2
RewriteRule USLUGE ?strana=3
RewriteRule REFERENCE ?strana=4
RewriteRule CENOVNIK ?strana=5
RewriteRule KONTAKT ?strana=6
ErrorDocument 404 /ivi-sistem/view/404.php
```

2.2.10.send.php

Slanjem poruke, odnosno klikom tastera 'Pošalji' na strani 'KONTAKT' aktivira se fajl send.php.

Ovaj fajl sadrži podatke koji su poslani putem HTTP zahteva 'post', a koji se u njemu nalaze kao vrednosti promenljivih. Te promenjive se zajedno sa e-mail adresom primaoca i naslovom poruke smeštaju u funkciju mail() koja se nalazi u kontroli toka. Kontrola toka proverava da li postoji aktivacija i da li je ispravno uneta e-mail adresa.

```
send.php
<?php
$name = $_POST['name'];
$email = $_POST['email'];
$post = $_POST['poruka'];
$send = $_POST['send'];

$to = "ivan.jojic81@gmail.com";
$subject = "Poruka sa mog sajta";

$message = "<b>Ime:</b> " . $name . "\r\n";
$message .= "<b>Email:</b> " . $email . "\r\n";
$message .= "<b>Poruka:</b> " . $post;

if (isset($send) ) {
    if (empty($email)) {
        echo "<h2>Greška, polje za Email je obavezno!</h2><form action='KONTAKT'><input type='submit' value='Pokušajte ponovo'></form>";
    }else if(!filter_var($email,FILTER_VALIDATE_EMAIL)){
        echo "<h2>Email adresa Vam je nepravna!</h2><form action='KONTAKT'><input type='submit' value='Pokušajte ponovo'></form>";
    }else{
        mail($to, $subject, $message);
        echo "<h1>Vaša poruka je poslana</h1><p>Zahvaljujemo na poslanoj poruci! Kontaktirati ćemo Vas u najkraćem mogućem roku.</p>";
    }
}
```

2.3.JavaScript

U fajlu myJavaScript.js nalazi se kod u okviru jQuery biblioteke koji omogućava da se slike na front delu strane naizmenično menjaju.

Na frontu stane u HTML kodu nalaze se tri slike, sa kojima preko CSS selektora JavaScript kod manipuliše.

Osim selektora, veći deo koda čini funkcija setInterval(), u kojoj se nalazi callback funkcija koja poziva samu sebe da se izvršava na svakih 4000 milisekundi.

Postoji mogućnost i podešavanja funkcija fadeIn() i fadeOut() za prikaz postepene providnosti, one se isto kao i setInterval() određuju po dužini trajanja u milisekundama.

Sistem funkcioniše tako da u HTML kodu se može naći neograničen broj slika.

```
myJavaScript.js
$(document).ready(function(){
    $("#header div:gt(0)").hide();
    setInterval(function() {
        $('#header div:first')
            .fadeOut(1500)
            .next()
            .fadeIn(1500)
            .end()
            .appendTo('#header');
    }, 4000);
});
```

```
<div id="header">
    <div>
        
    </div>
    <div>
        
    </div>
    <div>
        
    </div>
</div><!--end header-->
```