Web Tehnologije

Dokumentacija projektnih zadataka

Valjevčić Elvin

15561

1. Prva spirala

U prvoj spirali web stranica mi se sastoji od sljedeće 4 HTML datoteke, a to su: Naslovna.html, Onama.html, Kontakt.html i linkovi.html koje su povezane menijem tako da se sa svake podstranice može doći na bilo koju drugu podstranicu. Sve četiri pod stranice imaju unificiran dizajn i sve sadrže istu eksternu listu stilova CSS datoteku pod nazivom Stil.css i ona je uključena tako što sve četiri html datoteke sadrže vezu na nju u head sekciji: <link rel="stylesheet" href="Stil.css">

U HTML datotekama se ne nalazi style tag niti bilo koji tagovi koji definišu izgled stranice, sva definicija stila se nalazi u ovoj eksternoj css datoteci.

Sve 4 html i Stil.css datoteka su validne kroz [validator.w3.org](http://validator.w3.org/) i [jigsaw.w3.org/css-validator](http://jigsaw.w3.org/css-validator/) validatore.

Objašnjenje HTML-a

Sve četiri stranice imaju zajednički *<head>* dio, isto zaglavlje, lijevi dio i podnožje, a jedina im je razlika u glavnom dijelu *<div id="glavni\_dio">* tako da ćemo prvo opisati zajednički dio. Počinju na sljedeći način:

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="UTF-8">*

*<link rel="stylesheet" href="Stil.css">*

*<title> Stomatološka ordinacija 3D </title>*

*</head>*

Svi HTML dokumenti počinju sa deklaracijom *<!DOCTYPE html>* i sa *<html>* tagom, a završavaju sa *</html> tagom.* U head sekciji označeno je koja će se vrsta karaktera koristiti a to je *<meta charset="UTF-8">* sadrži vezu na eksternu datoteku Stil.css liste stila *<link rel="stylesheet" href="Stil.css">* i zadrži naslov stranice između tagova *<title> </title>*

Vidljivi dio HTML dokumenta je između tagova *<body>* i *</body>* i zatim slijedi zaglavlje:

*<body>*

*<div id="zaglavlje">*

*<img class="logo" src="logo.jpg" alt="logo">*

*<div id="meni">*

*<ul class="meni">*

*<li><a href="Naslovna.html">Novosti</a></li>*

*<li><a href="Onama.html">O nama</a></li>*

*<li><a href="Kontakt.html">Kontakt</a></li>*

*<li><a href="Linkovi.html">Linkovi</a></li>*

*</ul>*

*</div>*

*</div>*

U zaglavlju se nalazi logotip ordinacije koji je dodan sa *<img class="logo" src="logo.jpg" alt="logo">* src je izvor slike, a pošto se nalazi u istoj datoteci kao html stranice naveden je samo naziv logo.jpg. alt=“logo“ označava da se ispise logo ako iz nekog razloga nije prikazan. Nakon logotipa u zaglavlju se nalazi meni koji povezuje sve četiri pod stranice. Imlementiran je kao neuređena lista sa *<ul class="meni">* a svaka stavka liste *<li><a href="Naslovna.html">Novosti</a></li>* predstavlja link na HTML daoteku. Izgled menija je definisan u Stil.css datoteci sa id izbornikom #meni što će biti objašnjeno kada budemo objašnjavali sadržaj css datoteke.

Nakon zaglavlja slijedi lijevi dio stranice koji je također isti u sve četiri html datoteke:

*<div id="lijevo">*

*<img class="SlikaLocation" alt="lokacija" src="Pictures/location.png">*

*&nbsp; Branislava Đurđeva br.18*

*<p> 033 665-718 </p>*

*<p> Radno vrijeme: 08:00 - 18:00 </p>*

*<p> Posjetite našu fb stranicu:</p>*

*<a id="SlikaFb" target="a" href="https://www.facebook.com/pages/Stomatolo%C5%A1ka-ordinacija-3D/535914919867455?fref=ts">*

*<img alt="Facebook" src="https://beeinformed.org/wp-content/plugins/social-media-feather/synved-social/image/social/regular/96x96/facebook.png">*

*</a>*

*<div id="lokacija">*

*<p> Naša lokacija: </p>*

*<img src="http://imageprocessor.websimages.com/width/539/crop/42,0,462x346/stomatoloskaordinacija3d.webs.com/10456820\_535923326533281\_2923626144820835315\_n.jpg" alt="slika">*

*</div>*

*</div>*

U lijevom dijelu stranice prvo se nalaze podaci o ordinaciji: ulica, kontakt telefon, radno vrijeme. *<img class="SlikaLocation" alt="lokacija" src="Pictures/location.png">* predstavlja ikonu lokacije, nakon čega slijedi razmak sa ulicom: *&nbsp; Branislava Đurđeva br.18* &nbsp; predstavlja razmak u html-u. Inače gdje god je u tagovima navedeno class=“nešto“ to je radi definisanja izgleda u css datoteci. Nakon toga slijedi ikona facebooka čijim se klikom na nju otvara facebook stranica ordinacije:

*<a id="SlikaFb" target="a" href="https://www.facebook.com/pages/Stomatolo%C5%A1ka-ordinacija-3D/535914919867455?fref=ts">*

*<img alt="Facebook" src="https://beeinformed.org/wp-content/plugins/social-media-feather/synved-social/image/social/regular/96x96/facebook.png">*

*</a>*

*target="a"* znači da se klikom otvara novi prozor, u href= se nalazi link do fb stranice. U <img tagu je naveden alt ako se slika ne prikaže u src link slike a sve to se navodi između <a></a> tagova da predstavlja link.

Nakon toga u lijevom dijelu se nalazi slika lokacije gdje se ordinacija nalazi koja je dodata na vec objašnjen način sa <img> tagom.

Nakon lijevog dijela stranica sadrži sekciju *<div id="glavni\_dio">* koja se ralikuje u sve četiri podstranice tako da ćemo opisati sve četiri. Nakon glavnog djela slijedi podnožje koje je isto:

*<div id="podnozje"><p>Copyright &copy; Valjevčić Elvin 2015.</p></div>*

I na kraju se zatvaraju bodi i html tagovi:

*</body>*

*</html>*

Naslovna

Stranica koja je označena kao naslovna u sekciji glavni\_dio sadrži niz novosti. Svaka novost se sastoji od naslova, imena autora, datuma objave,slike, teksta i linka sa tekstom "Detaljnije..." koji za sada ne vodi nigdje kao što se i tražilo:

*<div id="glavni\_dio">*

*<div class="novost">*

*<h1 class="naslov"> Naslov 1</h1>*

*<p class="tekst\_novosti"> Ovo je neki tekst novosti </p>*

*<a href="#">Detaljnije...</a>*

*<br>*

*<img class="novosti\_slika" src="http://ikeda-dentalcl.com/wp-content/uploads/2014/02/image3.jpg" width="500" height="300" alt="slika">*

*<p class="datum"> Autor, Datum </p>*

*</div>*

*</div>*

Tag h1 označava naslov, ispod je tekst novosti i link detaljnije sa a tagom koji za sada ne vodi nigdje. <br> označava novi red u html-u, zati slijedi slika i autor i datum. Ovdje je dat samo primjer jedne novosti, a mogu se lahko dodavati ostale u sekciji *<div id="glavni\_dio">* kao sekcija *<div class="novost">,* tako da se stranica neće deformisati ma koliko novosti bilo.

Onama (u kojoj se nalazi tabela)

U Onama.html datoteci se u sekciji dlavnog dijela nalazi *<div id="onama">* sekcija koja sadrži tekst koji govori o ordinaciji i sadrži cjenovnik koji je prikazan kroz tabelu kao što se tražilo u zadatku.

*<div id="glavni\_dio">*

*<div id="onama">*

*<p>Stomatološka ordinacija 3D, na renomiranoj lokaciji, u Centru grada, nudi Vam sve usluge polivalentne stomatologije:</p>*

*<ul>*

*<li>Popravke zuba, sa helio i amalgamskim plombama</li>*

*<li>Vađenje zuba</li>*

*<li>Ultrazvucno skidanje kamenca i poliranje zuba</li>*

*<li>Izradu svih vrsta mobilnih i fiksnih protetskih radova; (keramickih i cirkonijum solo navlaka, nadogradnji, mostova, totalnih i parcijalnih proteza, kombinovanih radova itd.)</li>*

*<li>Izbjeljivanje zuba LED sistemom</li>*

*<li>Ugradnju estetskih detalja na zube ( ukrasni cirkon )</li>*

*</ul>*

*<p>Redovnim stomatoloskim pregledima, upotpunjenim dijagnostikom uz pomoc digitalnog snimka stanja cijelih usta i zuba:</p>*

*<ul>*

*<li>unapređujete svoje kompletno zdravlje</li>*

*<li>prevenirate nastanak, i kasno otkrivanje bilo kakvih promijena u kosti</li>*

*<li>imate uvid u stanje maksilarnog sinusa</li>*

*<li>imate uvid u stanje svih zuba, stanje starih plombi(postojanje karijesa ispod starih plombi)</li>*

*<li>imate uvid u mogucnost postojanja tzv.aproksimalnog karijesa,koji nije vidljiv klinickim pregledom u ustima</li>*

*<li>prevenirate nastanak „iznenadnog bola“, na zubima za koje ste mislili da nisu pokvareni, ili da su sanirani</li>*

*<li>smanjujete vrijeme provedeno kod stomatologa, koje inace zahtjevaju slozenije procedure</li>*

*<li>smanjujete svoje troskove</li>*

*</ul>*

Podaci su prikazani kroz neuređene liste i slijedi tabela :

*<table id="tabela“>*

*<caption>CJENOVNIK USLUGA</caption>*

*<tr>*

*<th>Usluga</th>*

*<th>Cijena</th>*

*<th>Popust</th>*

*</tr>*

*<tr>*

*<td>Stomatološki pregled</td>*

*<td class="sredina">20KM</td>*

*<td class="popust">-</td>*

*</tr>*

*<tr>*

*<td>Prva pomoć kod dentalgija</td>*

*<td class="sredina">20KM</td>*

*<td class="popust">-</td>*

*</tr>*

*<tr>*

*<td>Pečaćenje fisura (po zubu)</td>*

*<td class="sredina">30KM</td>*

*<td class="popust">-</td>*

*</tr>*

*<tr>*

*<td>Topikalna fluoridacija</td>*

*<td class="sredina">20KM</td>*

*<td class="popust">-</td>*

*</tr>*

*<tr>*

*<td>Lokalna i sprovoda anestezija</td>*

*<td class="sredina">10KM</td>*

*<td class="popust">-</td>*

*</tr>*

*</table>*

*<br>*

*</div>*

*</div>*

tabela se definiše tagom <table>. <caption> je naslov tabele. Između tagova <tr> i </tr> se nalazi red tabele u kojoj <td> predstavlja ćeliju u tom redu. Tabela ima tri kolone, a redovi se mogu lagano proširivati ubacivanjem <tr> tagova npr. :

*<tr>*

*<td>Lokalna i sprovoda anestezija</td>*

*<td class="sredina">10KM</td>*

*<td class="popust">-</td>*

*</tr>*

Linkovi

Na ovoj stranici se u sekciji glavnog dijela nalazi sekcija partneri koja pored naslova sadrži sekciju sa spiskom linkova na eksterne web stranice partnera ordinacije. Linkovi su predstavljeni u vidu neuređene liste (unordered list) gdje se umjesto tačke (bullet) prikazuje sličica:

*<div id="glavni\_dio">*

*<div id="partneri">*

*<h1>Partneri:</h1>*

*<ul class="linkovi">*

*<li><a href="http://www.citydeal.ba/">City deal</a></li>*

*<li><a href="http://www.3m.com">3M</a></li>*

*<li><a href="http://www.sirona.com/en/">Sirona</a></li>*

*</ul>*

*</div>*

*</div>*

tagovi <ul> predstavljaju neuređenu listu. U Stil.css datoteci za linkove se nalazi sljedeće:

*.linkovi {*

*list-style-image:url(*[*http://www.amrein.com/gif/arrowblueright.gif*](http://www.amrein.com/gif/arrowblueright.gif)*); }*

gdje se umjesto tačke prikazuje sličica čiji je link naveden u list-style-image:url()

Kontakt

U ovoj html datoteci je u sekciji glavni\_dio navedena kontakt forma za kontakt vlasnika stranice koja za sada nije funkcionalna:

*<div id="glavni\_dio">*

*<form id="kontakt\_forma">*

*<p>Ako se želite naručiti za pregled, postaviti neko pitanje ili dati neki svoj komentar to možete učiniti na formularu ispod:</p><br>*

*<p>Ime:&nbsp;*

*<input id="ime" name="ime" type="text">*

*<p>Prezime:&nbsp;*

*<input id="prezime" name="prezime" type="text">*

*<p>Mjesto:&nbsp;*

*<input id="mjesto" name="mjesto" type="text">*

*<p>Općina:&nbsp;*

*<input list="opcine" name="opcina">*

*<datalist id="opcine">*

*<option value="Novi Grad">*

*<option value="Novo Sarajevo">*

*<option value="Centar">*

*<option value="Stari Grad">*

*</datalist>*

*<p>E-mail: &nbsp;*

*<input id="email" name="email" type="text">*

*<p>Poruka:*

*<textarea id="poruka" name="poruka"></textarea> <br><br><br><br>*

*<input id="posalji" type="submit" value="Pošalji" onClick="Provjeri();">*

*Progres:<progress id="progres"form="kontakt\_forma" max=6 value=0></progress>*

*<video width="400" height="250" controls="controls" autoplay>*

*<source src="izbjeljivanje.mp4" type="video/mp4">*

*Vas browser ne podrzava video*

*</video>*

*</form>*

*</div>*

Forma se nalazi između tagova <form></form> sadrži polja za unos koji se definišu sa input tagom i atributom type=text i button za potvrdu slanja koji se definiše sa input tagom i atributom type=submit.

Od HTML5 kontrola koje se nalaze na ovoj stranici su video koji je ubačen sa ovim djelom:

<*video width="400" height="250" controls="controls" autoplay>*

*<source src="izbjeljivanje.mp4" type="video/mp4">*

*Vas browser ne podrzava video*

*</video>*

u atributu src se navodi link do videa, a pošto se nalazi u istom folderu kao i html stranica naveden je samo naziv izbjeljivanje.mp4 . Iduće HTML5 kontrole koje su ubačene su padajuća lista za općinu:

*<input id="opcina" list="opcine" name="opcina">*

*<datalist id="opcine">*

*<option value="Novi Grad">*

*<option value="Novo Sarajevo">*

*<option value="Centar">*

*<option value="Stari Grad">*

*</datalist>*

i progress bar:

*<progress id="progres"form="kontakt\_forma" max=6 value=0></progress>*

koji se prikazuju ispravno na vodećim web browserima.

Stil

Na početku Stil.css datoteke su definisana neka svojstva body sekcije njena boja pozadine, font i podrazumjevana boja slova:

*body {*

*background-color:green;*

*font-family:sans-serif;*

*color:green; }*

Nakon toga ide veličina loga, lijeva margina, bijela boja za zaglavlje i 10px razmaka između krajeva stranice i zaglavlja:

*.logo {*

*width: 450px;*

*height: 120px;*

*margin-left:20px; }*

*#zaglavlje {*

*background-color:white;*

*margin:10px; }*

U css datoteci kod koji definiše izgled gornjeg menija je sljedeći:

*#meni {*

*margin-top:30px;*

*float:right; }*

*#meni li{*

*display:inline-block; }*

*#meni li a{*

*text-decoration:none;*

*color:green;*

*border: medium green solid;*

*padding:5px;*

*font-size:18pt;*

*margin-right:30px;*

*box-shadow: 0px 0px 15px green; }*

#meni označava id izbornik tj. definiše se izgled <div id="meni">. margin-top:30px označava razmak od 30 piksela sa gornje strane, a float:right da meni pluta na desnoj strani stranice. display:inline-block označava da je meni prikazan kao vodoravan umjesto vertikalan. border: medium green solid označava okvir zelene boje font-size veličinu slova, color boju, padding razmak između slova i okvira itd. text-decoration:none; znači da ukidamo podrazumjevano podvlačenje ispod teksta.

Lijeva sekcija pluta u lijevo sa float:left, definisane su joj margine i veličina i boja slova:

*#lijevo {*

*float:left;*

*margin:10px;*

*margin-top:25px;*

*color:white;*

*font-size:14pt;*

*}*

Razmak fb slike u lijevoj sekciji: #SlikaFb {

padding:110px;

}

Glavni dio je obojen u bijelo sa zelenim slovima, a pomoću margina je određena njegova pozicija. Lijeva margina je veća od ostalih pošto se na lijevo strani nalazi lijevi dio:

*#glavni\_dio {*

*background-color:white;*

*color:green;*

*margin-top:30px;*

*margin-left:350px;*

*margin-right:10px;*

*margin-bottom:30px;*

*}*

Za svaku novost je postavljen okvir zelene boje koji ima sjenku:

*.novost {*

*border: medium green solid;*

*padding:15px;*

*box-shadow: 0px 0px 15px green;*

*}*

Osobine novosti, poput boje i poravnavanja, smanjen font za datum su definisani sljedečim:

*.naslov {*

*color:green;*

*text-align:left; }*

*p.tekst\_novosti {*

*text-align: justify; }*

*.novosti\_slika{*

*margin:5px;*

*margin-left:auto;*

*margin-right:auto; }*

*a{*

*color:green;}*

*.datum{*

*font-size:small; }*

Kod partnera također imamo okvir, sjenku, razmake:

*#partneri {*

*padding:20px;*

*border: medium green solid;*

*box-shadow: 0px 0px 15px green;*

*height:600px;*

*}*

*#partneri li {*

*padding:15px;*

*font-size:17pt;*

*margin-left:40px;*

*}*

a sličice umjesto tačke su dodane na sljedeći način:

*.linkovi {*

*list-style-image:url(*[*http://www.amrein.com/gif/arrowblueright.gif)*](http://www.amrein.com/gif/arrowblueright.gif))*;}*

gdje je u zagradi naveden link upotrijebljene sličice.

Za onama također isti okvir i sjenka:

*#onama {*

*padding: 15px;*

*border: medium green solid;*

*box-shadow: 0px 0px 15px green;*

*}*

Za tabelu je definisano sljedeće:

*#tabela{*

*margin:50px;*

*width:90%;*

*border: thin green solid;*

*box-shadow: 0px 0px 15px green; }*

*#tabela .sredina {*

*text-align:center;}*

*#tabela .popust{*

*text-align:center; }*

gdje width:90% predstavlja da širina tabele zauzima 90 posto stranice, border je okvir tabele box-shadow sjenka. Sa *#tabela .sredina* i *#tabela .popust* definišemo poravnanje teksta u srednjoj koloni i u koloni popust na srednje, dok prva kolona ostaje defaultno poravnata na lijevo.

Izgled kontakt forme je definisan sljedećim gdje se se najviše podešavale pozicije pojedinih elemenata na formi:

*#kontakt\_forma {*

*padding:60px;*

*border: medium green solid;*

*box-shadow: 0px 0px 15px green;*

*height:600px; }*

*video {*

*float:right; }*

*#kontakt\_forma #progres {*

*position:absolute;*

*width:185px;*

*left:560px; }*

*#kontakt\_forma input {*

*position:absolute;*

*left:560px;*

*width:180px; }*

*#kontakt\_forma textarea {*

*position:absolute;*

*left:560px;*

*width:180px;*

*height:65px; }*

*#kontakt\_forma #posalji {*

*left:420px;*

*top:630px;*

*color:green;*

*width:105px; }*

I na kraju je podešeno podnožje, boja, font i boja teksta, razmaci:

*#podnozje {*

*background-color:white;*

*margin:10px;*

*color:green;*

*font-size:small;*

*text-align:center;*

*padding:0.1px;*

*}*

2. Druga spirala

Ono što je dodano novo u ovoj spirali je validacija kontakt forme kroz java script i meni.

Validacija forme

U ovoj spirali je rađena validacija forme kroz javaScript. U svim html datotekama je dodana veza na eksternu javaScript datoteku koja sadrži funkciju koja validira formu koja će biti objašnjena i funkciju za prikaz i sakrivanje menija koja će također biti objašnjena. Datoteka nosi naziv Validacija2.js i uključena je u html datotekama prije </body> taga na sljedeći način: <script src="Validacija2.js"></script>

Validacija2.js sadrži funkciju koja služi za validiranje forme koja nosi naziv provjeriFormu() koja se poziva onog trenutka kada pošaljemo formu a to smo definisali tako što smo u <input id="posalji" type="submit" value="Pošalji"> dodali onclick event da se klikom na slanje forme poziva funkcija za validiranje forme: <input id="posalji" type="submit" value="Pošalji" onclick=" return provjeriFormu()">

Ono što je dodano na našoj kontakt formi je sekcija gdje će se ipisivati greška:

*<form id="kontakt\_forma">*

*<p>Ako se želite naručiti za pregled, postaviti neko pitanje ili dati neki svoj komentar to možete učiniti na formularu ispod:</p><br>*

*<div id="greska"></div>*

Također smo u formi kod svakog elementa dodali sličice upozorenja sa uzvičnikom :

*<p>\*Ime:&nbsp;*

*<input id="ime" name="ime" type="text">*

*<img class="upozorenje" id="slika\_ime" src="Pictures/warning.png" alt="slika\_ime">*

*<p>\*Prezime:&nbsp;*

*<input id="prezime" name="prezime" type="text">*

*<img class="upozorenje" id="slika\_prezime" src="Pictures/warning.png" alt="slika\_prezime">*

*...*

koje su u Stil.css sakrivene sa displey:none, a prikazuju se samo u slučaju da postoji greška tako da je u javaScript funkciji dodano da se prikazuju za odgovorajuću grešku na sljedeći način: document.getElementById('slika\_ime').style.display="block";

U Stil.css datoteci je dodano:

*.upozorenje {*

*position:absolute;*

*left:760px;*

*margin-top:-15px;*

*display:none; }*

*#greska {*

*color:red;*

*font-weight:bold; }*

gdje se podešava lokacija sličica upozorenja da budu tačno pored odgovarajućeg elementa i podešava se da budu nevidljive i određeno je da slova u sekciji gdje će se ispisivati greška budu boldirana i crvene boje.

Funkcija koja se nalazi u Validacija2.js provjeriFormu() glasi i radi na sljedeći način:

*function provjeriFormu() {*

*var greska = document.getElementById('greska');*

*greska.innerHTML="";*

*for(var i=0;i<5;i++)*

*document.getElementsByClassName('upozorenje')[i].style.display="none";*

Varijabli greska pridružujemo div sekciju gdje se ispisuje greška i čistimo predhodne greške. For petljom sve sličice upozorenja sakrivamo za slučaj da se neka prije pokazala.

*var forma=document.getElementById('kontakt\_forma');*

*if (forma.ime.value.length>10 || forma.ime.value.length<3) {*

*greska.innerHTML+="Predugo/prekratko ime<br>";*

*document.getElementById('slika\_ime').style.display="block";*

*return false;*

*}*

*if (forma.prezime.value.length>16 || forma.prezime.value.length<3) {*

*greska.innerHTML+="Predugo/prekratko prezime<br>";*

*document.getElementById('slika\_prezime').style.display="block";*

*return false;*

*}*

*if (forma.email.value.length==0) {*

*greska.innerHTML+="Molimo unesite E-mail";*

*document.getElementById('slika\_email').style.display="block";*

*return false;*

*}*

Dalje u varijablu forma smještamo našu kontakt formu i provjeravamo da li je ime dužine između 3 i 10 znakova, ako nije ispisuje se odgovarajuća greška sa *greska.innerHTML+="Predugo/prekratko ime<br>";* , prikazuje se odgovarajuća sličica upozorenja sa *document.getElementById('slika\_ime').style.display="block";* i funkcija vraća false zbog toga da se forma ne bi mogla poslati.

Analogno je urađeno i ako prezime nije odgovarajuće dužine i ako je polje za email prazno. Preostali dio funkcije je sljedeći:

*var mailRegEx = /^.+@[^\.].\*\.[a-z]{2,}$/;*

*if (!mailRegEx.test(forma.email.value)) {*

*greska.innerHTML+="Neispravan format E-maila";*

*document.getElementById('slika\_email').style.display="block";*

*return false;*

*}*

*if (forma.email.value != forma.email2.value) {*

*greska.innerHTML+="E-malovi se ne podudaraju";*

*document.getElementById('slika\_email').style.display="block";*

*document.getElementById('slika\_email2').style.display="block";*

*return false;*

*}*

*return true;*

*}*

gdje se provjeravao format emaila pomoću regexa. /^.+ označava bilo koji niz znakova i da mora biti barem jedan znak prije @ uglaste zagrade označavaju bilo koji elemenat unutar njih [a-z] bilo koji znak između a i z {2,} označava najmanje dva znaka između a i z.

Polja na formi koja su međusobno zavisna jedna od drugog su email adrese, tj. pošto se traži da se potvrdi email adresa obje moraju biti jednake što se provjeravalo sa : *if (forma.email.value != forma.email2.value)*

Meni

Padajući meni funkcioniše po principu menija u dekstop aplikacijama. Kod mene je implementiran tako da kada se klikne na meni O nama otvori se pod meni sa još dva linka a to su Informacije i Proizvodi (koji će biti implementirani u trećoj spirali, a za sada ne vodi nigdje). Implementirano je tako što je u svim html datotekama kod menija dodan pod meni na sljedeći način:

*<div id="meni">*

*<ul class="meni">*

*<li><a href="Naslovna.html">Novosti</a></li>*

*<li><a href="Onama.html">O nama</a></li>*

*<li><a id="o\_nama" onclick="prikaziPodMeni()">O nama</a>*

*<ul id="podmeni">*

*<li><a href="Onama.html">Informacije</a></li>*

*<li><a href="#">Proizvodi</a></li>*

*</ul>*

*</li>*

*<li><a href="Kontakt.html">Kontakt</a></li>*

*<li><a href="Linkovi.html">Linkovi</a></li>*

*</ul>*

*</div>*

kod linka O nama je dodan onclick event koji klikom na njega poziva funkciju koja se nalazi u javaScript datoteci prikaziPodMeni() koja prikazuje i sakriva pod meni koji je naveden u sekciji ispod O nama

*<ul id="podmeni">*

*<li><a href="Onama.html">Informacije</a></li>*

*<li><a href="# ">Proizvodi</a></li>*

*</ul>*

Ovaj pod meni je sakriven tako što je u Stil.css navedeno visibility: hidden; a otkriva se samo u slučaju da se klikne na Onama kada se poziva funkcija prikaziPodMeni() koja provjerava da li je podmeni vidljiv, ako nije prikazuje ga sa podmeni.style.visibility = "visible"; i postavlja sivu boju Onama elementa, a ako je meni vidljiv sakriva ga i isključuju sivu boju Onama elementa na prirodnu bijelu:

*function prikaziPodMeni() {*

*var podmeni = document.getElementById('podmeni');*

*var o\_nama = document.getElementById('o\_nama');*

*if(podmeni.style.visibility == "visible") {*

*podmeni.style.visibility = "hidden";*

*o\_nama.style.backgroundColor="white";*

*}*

*else {*

*podmeni.style.visibility = "visible";*

*o\_nama.style.backgroundColor="silver";*

*}*

*}*

Također kada se pređe mišem na neku opciju u podmeniju ona promjeni boju na sivu što je rješeno tako što je u Stil.css preko hover eventa navedeno:

*#meni ul li ul li a:hover {*

*background-color: silver;*

*}*

Znači za potrebe pod menija u css datoteci dodan je sljedeći kod:

*#meni ul li ul {*

*margin: 0;*

*padding:0;*

*margin-top: 20pt;*

*width: 120px;*

*position: absolute;*

*visibility: hidden; }*

*#meni ul li ul li {*

*margin-top:20px;*

*background-color:white;*

*height:50px; }*

*#meni ul li ul li a:hover {*

*background-color: silver; }*

Pored onih osobina koje su objašnjene iznad ostale osobine u ovom css djelu smještaju pod meni ispod O nama dijela.

3. Treća spirala

Validacija forme putem AJAXa

Na formi je dodana validacija putem ajaxa tako što se validiraju dva polja mjesto i općina pomoću web servisa koji je dat u postavku spirale. Kod unosa opcine i mjesta na kontakt formi je dodan onblur event koji poziva funkciju ProvjeriOpcinuMjesto():

*<input id="mjesto" name="mjesto" type="text" onblur="return ProvjeriOpcinuMjesto()">*

i

*<input id="opcina" list="opcine" name="opcina" onblur="return ProvjeriOpcinuMjesto()">*

Validira se da li postoji mjesto, da li postoji općina i da li je mjesto u datoj općini. Validacija je urađena u funkciji ProvjeriOpcinuMjesto() koja se nalazi u javaScript datoteci Validacija2.js. Funkcija glasi ovako:

*var x=0;*

*function ProvjeriOpcinuMjesto()*

*{*

*var mjesto = document.getElementById("mjesto").value;*

*var opcina = document.getElementById("opcina").value;*

*if (mjesto=="" || opcina=="") {return false;}*

*else {*

*var xmlhttp=new XMLHttpRequest();*

*xmlhttp.onreadystatechange=function() {*

*if (xmlhttp.readyState==4 && xmlhttp.status==200) {*

*var greska2 = JSON.parse(xmlhttp.responseText);*

*if (greska2.greska!=undefined) {*

*x=1;*

*document.getElementById('greska').innerHTML=greska2.greska;*

*document.getElementById('slika\_opcina').style.display="block"; document.getElementById('slika\_mjesto').style.display="block";*

*}*

*else {*

*x=0; document.getElementById('greska').innerHTML=greska2.ok;*

*document.getElementById('slika\_opcina').style.display="none"; document.getElementById('slika\_mjesto').style.display="none";*

*}*

*}*

*}*

*mjesto = encodeURIComponent(mjesto);*

*opcina = encodeURIComponent(opcina); xmlhttp.open("GET","http://zamger.etf.unsa.ba/wt/mjesto\_opcina.php?opcina="+opcina+"&mjesto="+mjesto,true);*

*xmlhttp.send();*

*return false;*

*}*

*}*

Ova funkcija radi na sljedeći način. Prvo se provjerava da li su mjesto i općina prazni, ako jesu funkcija vraća false i ne radi ništa, a ako nisu nastavlja sa radom. Ajax je realiziran preko XMLhttpRequest tako da se on kreira. Ključna metoda ovog objekta je open:

*xmlhttp.open("GET","http://zamger.etf.unsa.ba/wt/mjesto\_opcina.php?opcina="+opcina+"&mjesto="+mjesto,true);*

koji u našem slučaju Get metodom šalje zahtjev web servisu sa parametrima opcina i mjesto sa naše forme koji su navedeni poslije linka web servisa poslije znaka ? i svaki idući parametar se navodi znakom &. Treći parametar atributa je true što znači da je poziv asihron. Poziv se upućuje metodom send(). Pošto je poziv asihron definisana je funkcija koja se izvršava kada se dobije odgovor od servera a to je funkcija koja se pridružuje atributu onreadystatechange.Uslov *if (xmlhttp.readyState==4 && xmlhttp.status==200)* prolazi u slučaju ako je uspješno završen zahtjev. Zatim pošto server vraća JSON objekat potrebno ga je parsirati sa

*var greska2 = JSON.parse(xmlhttp.responseText);*

sa *if (greska2.greska!=undefined)* provjeravamo da li je varijabla greska2 uopste greska ili je potvrda da je mjesto u datoj općini, ako je greška varijabla x se postavlja na jedinicu kako bi znali da se forma ne može poslati, a ako nije varijabla x je nula i forma se može poslati uz ispisanu poruku sa web servera da je mjesto iz date općine.

Ova funkcija se poziva u eventima onblur na unosima mjesta i općine koja se poziva odpuštanjem fokusa sa datog elementa i pošto se ajax poziva asihrono odmah se ispisuje ima li greške ili ne.

Također pri slanju forme u funkciji provjeriFormu() se provjerava varijabla x tj. ima li greške, ako ima forma se ne može poslati tako da je u toj funkciji dodano sljedeće:

*//ajax za mjesto i opcinu*

*if (x==1) {*

*greska.innerHTML+="Unesite odgovarajuće mjesto i općinu";*

*document.getElementById('slika\_mjesto').style.display="block";*

*document.getElementById('slika\_opcina').style.display="block";*

*return false;*

*}*

Učitavanje podstranica putem AJAXa

Ovdje je cilj bio da stranica postane single page application tj. da se prelaskom između podstranica ne vrši prebacivanje nego da se učitavaju putem ajaxa asinhrono. To je urađeno tako što su u svim html datotekama u meniju linkovi zamjenjeni sa pozivom funkcije ucitaj(adresa) na sljedeći način:

*<div id="meni">*

*<ul class="meni">*

*<li><a onclick="return ucitaj('Naslovna.php')">Novosti</a></li>*

*<li><a id="o\_nama" onclick="prikaziPodMeni()">O nama</a>*

*<ul id="podmeni">*

*<li><a onclick="return ucitaj('Onama.html')">Informacije</a></li>*

*<li><a onclick="return ucitaj2('Proizvodi.html')">Proizvodi</a></li>*

*</ul>*

*</li>*

*<li><a onclick="return ucitaj('Kontakt.html')">Kontakt</a></li>*

*<li><a onclick="return ucitaj('Linkovi.html')">Linkovi</a></li>*

*</ul>*

*</div>*

Funkcija i html datoteka *ucitaj2('Proizvodi.html')* će biti navedene i objašnjene kasnije, a funkcija ucitaj(adresa) je implementirana u javaScript datoteci na sljedeći način:

*function ucitaj(adresa) {*

*var ajax = new XMLHttpRequest();*

*ajax.onreadystatechange = function () {*

*if (ajax.readyState == 4 && ajax.status == 200) {*

*document.body.innerHTML = ajax.responseText;*

*}*

*if (ajax.readyState == 4 && ajax.status == 404)*

*document.body.innerHTML = "Greska: nepoznat URL";*

*}*

*ajax.open("GET", adresa, true);*

*ajax.send();*

*return false;*

*}*

sa document.body.innerHTML= ajax.responseText; se učitava u body sekciju odgovor sa servera a to je naša stranica koja se prikazuje asinhrono.

Rad sa tabelom putem AJAXa (Proizvodi.hmtl)

U ovom djelu je implementirana html stranica Proizvodi.html koja pored onoga što sadrže sve ostale podstranice sadrži u glavnoj sekciji formu za unos, izmjenu i brisanje proizvoda i sekciju *<div id="proizvodi"> </div>*gdje će se prikazivati proizvodi kroz javaScript funkciju:

*<div id="glavni\_dio">*

*<form id="formaUnosProizvoda">*

*<p>Unesite podatke od proizvoda:</p><br>*

*<p>\*ID:&nbsp;*

*<input id="idP" class="unos" name="idP" type="text">*

*<p>\*Naziv:&nbsp;*

*<input id="nazivP" class="unos" name="nazivP" type="text">*

*<p>\*URL Slike:&nbsp;*

*<input id="urlSlike" class="unos" name="urlSlike" type="text">*

*<p>Opis:*

*<textarea id="opis" class="unos" name="opis"></textarea> <br><br><br><br>*

*<p> \*Obavezna polja </p>*

*<input id="Dodaj" type="submit" value="Dodaj proizvod" onclick="return dodajProizvod()">*

*<input id="Izmjeni" type="submit" value="Izmjeni proizvod" onclick="return izmijeniProizvod()">*

*<input id="Obrisi" type="submit" value="Obriši proizvod" onclick="return obrisiProizvod()">*

*</form>*

*<div id="proizvodi">*

*</div>*

*</div>*

Na kraju ove stranice je uključena javaScript datoteka Proizvodi.js *<script src="Proizvodi.js"></script>* u kojoj su implementirane funkcije za prikazivanje, dodavanje, izmjenu i brisanje proizvoda koje se pozivaju na onclick event odgovarajućih submint buttona. Funkcija PrikažiProizvode() izgleda ovako:

*function prikaziProizvode() {*

*var ajax;*

*ajax=new XMLHttpRequest();*

*ajax.onreadystatechange = function()*

*{*

*if (ajax.readyState == 4 && ajax.status == 200 ){*

*preuzmi(ajax.responseText);*

*}*

*}*

*ajax.open("POST","http://zamger.etf.unsa.ba/wt/proizvodi.php?brindexa=15561", true);*

*ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");*

*ajax.send();*

*}*

Radi tako što se šalje ajax zahtjev post metodom web servisu sa parametrom brojem indeksa:

*ajax.open("POST","http://zamger.etf.unsa.ba/wt/proizvodi.php?brindexa=15561", true);*

*ajax.send();*

U funkciji koja se poziva kada dođe odgovor sa servera preuzimaju se podaci o proizvodima sa funkcijom *preuzmi(ajax.responseText);* koja glasi ovako:

*function preuzmi(response)*

*{*

*var niz= JSON.parse(response);*

*var izlaz='<table id="tabela" border="1"> <caption>Proizvodi</caption>';*

*izlaz=izlaz+ "<tr><th>ID</th><th>Naziv</th><th>Opis</th><th>Slika</th><tr>";*

*for(var i=0; i < niz.length; i++)*

*{*

*var id=niz[i].id;*

*var naziv=niz[i].naziv;*

*var slika=niz[i].slika;*

*var opis=niz[i].opis;*

*izlaz=izlaz+ "<tr><td>"+id + "</td>"+"<td>"+naziv + "</td><td>";*

*izlaz= izlaz + opis +"</td><td><img src=" + slika + " alt='' height='150px' weight='230px'></td></tr>";*

*}*

*izlaz= izlaz +'</table><br>';*

*document.getElementById("proizvodi").innerHTML = izlaz;*

*}*

u kojoj se podacii smještaju u tabelu u varijablu izlaz koja se na kraju smješta u html sekciju proizvodi na sljedeći način:

*document.getElementById("proizvodi").innerHTML = izlaz;*

Ranije smo primjetili da kod učitavanja Proizvodi.html se poziva funkcija *ucitaj2(Proizvodi.html)* koja pored toga što učita Proizvodi.html preko ajaxa što je prije objašnjeno ona još poziva funkciju prikaziProizvode() koja je gore također objašnjena da bi se automatski prikazali proizvodi.

Sljedeća funkcija koja je implementirana u Proizvodi.js je dodajProizvod() koja izgleda i radi na sljedeći način:

*function dodajProizvod() {*

*var forma = document.getElementById('formaUnosProizvoda');*

*var naziv = forma.nazivP.value;*

*var opis = forma.opis.value;*

*var slika = forma.urlSlike.value;*

*var proizvod = {*

*naziv: naziv,*

*opis: opis,*

*slika: slika,*

*};*

*var ajax=new XMLHttpRequest();*

*ajax.onreadystatechange=function(){*

*if(ajax.status === 200 & ajax.readyState === 4)*

*{*

*alert("Dodali ste proizvod!");*

*prikaziProizvode();*

*}*

*}*

*ajax.open("POST", "http://zamger.etf.unsa.ba/wt/proizvodi.php?brindexa=15561", true);*

*ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");*

*ajax.send("akcija=dodavanje" + "&brindexa=15561&proizvod=" + JSON.stringify(proizvod));*

*return false;*

*}*

Prvo kupi podatke o proizvodu koji su uneseni preko forme. Onda se ti podaci šalju web servisu Post metodom sa parametrima akcija=dodavanje, brojem indeksa i proizvodom koji želimo unijeti:

*ajax.open("POST", "http://zamger.etf.unsa.ba/wt/proizvodi.php?brindexa=15561", true);*

*ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");*

*ajax.send("akcija=dodavanje" + "&brindexa=15561&proizvod=" + JSON.stringify(proizvod));*

Kada se unese proizvod poziva se funkcija koja je definisana u ajax.onreadystatechange i parametri *if(ajax.status === 200 & ajax.readyState === 4)* označavaju da je proizvod uspješno dodan što ispisujemo alert("Dodali ste proizvod!");

i poziva se funkcija koju smo predhodno objasnili prikaziProizvode(); koja također putem ajaxa prikazuje što znači da čim preko forme dodamo proizvod automatski će se prikazati u tabeli bez učitavanja stranice.

Ekvivalentno rade i preostale dvije funkcije izmijeniProizvod() i obrisiProizvod() samo što se šalju drugi parametri za izmjeni:

*ajax.send("akcija=promjena" + "&brindexa=15561&proizvod=" + JSON.stringify(proizvod));*

a za brisanje:

*ajax.send("akcija=brisanje" + "&brindexa=15561&proizvod=" + JSON.stringify(proizvod));*

U ovoj spirali je još dodano u Stil.css datoteku sljedeće osobine koje definišu poziciju forme za unos proizvoda:

*#formaUnosProizvoda {*

*padding:60px;*

*}*

*#formaUnosProizvoda .unos{*

*position:absolute;*

*left:540px;*

*}*

4. Četvrta spirala

U ovoj spirali je isključena validacije preko java script funkcije, a impelentirana je validacija korz php skriptu koja nosi naziv php\_validacija.php. Da bi smo to omogućili dodano je tagu naše kontakt formu sljedeće:

*form id="kontakt\_forma" action="php\_validacija.php" method="get">*

Što znači da se slanjem forme poziva php skripta koja preuzima podatke sa forme get metodom sa komandom $\_GET["ime ulaza"] i validira ih. Na početku skripte se nalazi sljedeće:

*<?php*

*$imeErr="";*

*$ime = $prezime= $email=$forma = $email2=$mjesto=$opcina=$poruka= "";*

*function test\_input($data) {*

*$data = trim($data);*

*$data = stripslashes($data);*

*$data = htmlspecialchars($data);*

*return $data;*

*}*

<?php označava početak php skripte, deklarisane su varijable koje će nam trebati, a zatim je imlementirana funkcija test\_input koja koja će se pozivati nad svim varijablama u formi koja služi protiv XSS propista. Dalje:

*if(isset($\_GET["slanje"])) {*

*$validacija = true;*

*if (empty($\_GET["ime"])) {*

*$imeErr = "Ime ne smije biti prazno!";*

*$validacija = false;*

*} else {*

*$ime = test\_input($\_GET["ime"]);*

*// da li sadrži samo slova i razmake*

*if (!preg\_match("/^[a-zA-Z ]\*$/",$ime)) {*

*$imeErr = "Samo slova i razmaci su dozvoljeni!";*

*$validacija = false;*

*}*

*}*

Sa if(isset($\_GET["slanje"])) provjerava da li je pritisnuto dugmo za slanje, ako jeste varijabla $validacija se postavlja na true, ako validacije na bude prolazila ta varijabla će biti postavljena na false. Zatim se validira polje ime tj. da li je prazno i da li sadrži samo slova i razmake ako ne prolazi varijabla $imeErr poprima vrijednost odgovarajuće greške i varijabla $validacija se postavlja na false. U nastavku skripte analogno se provjerava i prezime. Za email se provjerava format kao i kod javaScript validacije:

*$email = test\_input($\_GET["email"]);*

*if (!preg\_match("/^.+@[^\.].\*\.[a-z]{2,}$/",$email)) {*

*$imeErr = "Netačan email format!";*

*$validacija = false*

*}*

*i provjerava se da li se emailovi podudaraju:*

*if($email2 != $email) {*

*$imeErr = "Email-ovi moraju biti isti!";*

*$validacija = false;*

*}*

U nastavku kada smo validirali polja provjerava se da li je varijabla $validacija true ili false tj. da li validacija prolazi ili ne prolazi. U slučaju da ne prolazi poziva se ista ova forma sa greškama i sa istim vrijednostima koje je korisnik unio a to se rješava tako što se kod elemenata doda u value atribut ona vrijednost koja je predhono poslata npr.:

za ispis greške: *<div id="greska">greska: <?php print $imeErr; ?></div>*

da bude popunjena sa istim vrijednostima:

*<input id="ime" name="ime" type="text" value="<?php echo $ime; ?>">*

U slučaju da validacija prolazi tj. ako je varijavla $validacija==true izvršava se datoteka potvrda.php koja je uključena sa include naredbom:

*if($validacija==true) {*

*include 'potvrda.php';*

*}*

potvrda.php sadrži što sadrže sve ostale html stranice (zaglavlje,meni...) sa tim što u sekciji glavni\_dio sadrži sekciju *<div id="potvrda">* u kojoj se nalazi ono što se tražilo u postavci spirale.

*<div id="glavni\_dio">*

*<div id="potvrda">*

*<h3>Provjerite da li ste ispravno popunili kontakt formu</h3>*

*<br>*

*<?php print "Ime: " .$ime ?>*

*<br><br>*

*<?php print "Prezime: " .$prezime ?>*

*<br><br>*

*<?php print "Mjesto: " .$mjesto ?>*

*<br><br>*

*<?php print "Opcina: " .$opcina ?>*

*<br><br>*

*<?php print "Email: " .$email ?>*

*<br><br>*

*<?php print "Email2: " .$email2 ?>*

*<br><br>*

*<?php print "Poruka: " .$poruka ?>*

*<br><br>*

Prvo se prikazuju uneseni elementi. Zatim se ispisuje da li je korisnik siguran i forma sa buttonom „Siguran sam“ako jeste. Na njoj se nalaze skriveni ulazi kako bi se get metodom poslali podaci skripti posalji\_mail.php :

*<h3>Da li ste sigurni da želite poslati ove podatke?</h3>*

*<form method="get" name = "mail" action="posalji\_mail.php">*

*<input type="submit" name="send" value="Siguran sam">*

*<input type="hidden" name="ime" value="<?php echo $ime;?>">*

*<input type="hidden" name="email" value="<?php echo $email;?>">*

*<input type="hidden" name="poruka" value="<?php echo $poruka;?>">*

*</form>*

I ispod ovoga se nalazi forma za ponovno slanje sa unesenim predhodnim podacima ako korisnik želi nešto da izmjeni koja je ista kao ranije pa je nećemo ponavljati čijim se slanjem opet poziva skripta za validiranje i tako u krug sve dok korisnik ne pritisne dugme „Siguran sam“ kojim će se poslati email. U skripti posalji\_mail.php se nalazi sljedeće i da bi radilo potrebno je imati instaliran mail server:

*<?php*

*ini\_set("SMTP", "webmail.etf.unsa.ba");*

*ini\_set("smtp\_port", 25);*

*ini\_set("sendmail\_from","evaljevcic1@etf.unsa.ba");*

*$ime = $\_GET["ime"];*

*$Email = $\_GET["email"];*

*$poruka = $\_GET["poruka"];*

*$sub = "Proba";*

*$Ime = $ime; //senders name*

*$email = $Email; //senders e-mail adress*

*$recipient = "evaljevcic1@etf.unsa.ba"; //recipient*

*$mail\_body = $poruka; //mail body*

*$subject = $sub; //subject*

*$header = "From: ". $Ime . " <" . $email . ">\r\n"; //optional headerfields*

*if (mail($recipient, $subject, $mail\_body))*

*echo "Zahvaljujemo se što ste nas kontaktirali";*

*else*

*echo "Poruka nije poslana!"*

*?>*

Novosti

U ovom djelu je omogućeno da se novosti prikazuju iz TXT datoteke. To je omogućeno tako što je u Naslovna.html dodano u djelu gdje se prikazuju novosti sljedeće:

<div id="novosti">

<?php include "NovostiIzDat.php"; ?>

</div>

tj. uključuje se php skripta koja čita i prikazuje novosti iz datoteke koja će biti objašnjena u nastavku. Samim tim što sada Naslovna.html uključuje php kod treba prepraviti u Naslovna.php, pa i u meniju na svakom mjesto je prepravljono da se učitava Naslovna.php

U istom folderu gdje se nalazi php skripta se nalazi folder pod imenom novosti u kojoj se nalaze txt fajlovi sa formatom koji je naveden u spirali. Zbog toga na početku NovostiIzDat.php se nalazi sljedeći kod:

*<?php*

*$counter=0;*

*$novosti = array();*

*foreach(glob("novosti/\*.txt") as $imeFajla)*

*{*

*$novosti[$counter] = file($imeFajla);*

*$counter++;*

*}*

*$kolicinaNovosti=count($novosti);*

gdje se u niz $novosti smještaju novosti iz svih txt fajlova. Sljedećim kodom se postavlja format vremena novosti u nizu $novosti i sortiraju se po datumu i vremenu objavljivanja:

*for ($i=0; $i<$kolicinaNovosti; $i++)*

*{*

*for ($j=0; $j<$kolicinaNovosti-1-$i; $j++) {*

*$time1 = strtotime($novosti[$j][0]); $newformat1 = date('d-m-Y h:i:s',$time1);*

*$time2 = strtotime($novosti[$j+1][0]); $newformat2 = date('d-m-Y h:i:s',$time2);*

*if ($time2 > $time1) {*

*$tmp=$novosti[$j];*

*$novosti[$j]=$novosti[$j+1];*

*$novosti[$j+1]=$tmp;*

*}*

*}*

*}*

Dalje se u petlji for ($i=0; $i<$kolicinaNovosti; $i++) uzimaju podaci datum, autor, naslov i slika na sljedeći način:

*for ($i=0; $i<$kolicinaNovosti; $i++)*

*{*

*$novostLength=count($novosti[$i]);*

*$sadrzajNovosti=$detaljnijeNovosti="";*

*$datum=$novosti[$i][0]; $autor=$novosti[$i][1]; $naslov=$novosti[$i][2]; $slika=$novosti[$i][3];*

Dalje se traže znakovi „--“ i nakon njih se uzimaju detaljnije novosti u varijablu $detaljnijeNovosti, a prije toga sadržaj novosti u varijablu $$sadrzajNovosti sljedećim kodom:

*$j=4;*

*while ($j<$novostLength){*

*if (trim($novosti[$i][$j])=="--"){*

*for ($k=$j+1; $k<$novostLength; $k++){*

*$detaljnijeNovosti.=$novosti[$i][$k];*

*}*

*break;*

*}*

*$sadrzajNovosti.=$novosti[$i][$j];*

*$j++;*

*}*

varijbla $j je postavljena na 4 iz tog razloga jer se tekst novosti nalazi od petog reda.

$naslov = ucfirst(strtolower($naslov)); prebacuje ostala slova naslova osim prvog u mala.

Dalje se u skripti prikazuje html kod novosti kao i prije sa varijablama koje su očitane iz datoteke:

*div class="novost">*

*<h1 class="naslov"><?php echo htmlspecialchars(trim($naslov), ENT\_QUOTES, 'UTF-8'); ?></h1>*

*<p class="tekst\_novosti"> <?php echo trim($sadrzajNovosti); ?></p>*

Funkcija trim služi za uklanjanje praznog prostora na krajevima stringa, a htmlspecialchars pretvara html da se ne prikazuje i postavlja string na UTF-8 u našem slučaju.

Nakon što se prikažu naslov i tekst novosti dalje skripta ispituje da li postoji detaljni tekst novosti. Ako postoji pokazuje se link Detaljnije čijim klikom se otvara funkcija PrikaziPunuNovost koja se nalazi u javaScript datoteci. To je urađeno sljedećim kodom koji je u nastavku skripte:

*<?php if($detaljnijeNovosti!="") { ?>*

*<a onclick="return PrikaziPunuNovost('<?php echo htmlspecialchars(trim($datum), ENT\_QUOTES, 'UTF-8');?>','<?php echo htmlspecialchars(trim($naslov), ENT\_QUOTES, 'UTF-8');?>','<?php echo htmlspecialchars(trim($autor), ENT\_QUOTES, 'UTF-8');?>','<?php echo htmlspecialchars(trim($slika), ENT\_QUOTES, 'UTF-8');?>','<?php echo htmlspecialchars(trim($sadrzajNovosti), ENT\_QUOTES, 'UTF-8');?>','<?php echo htmlspecialchars(trim($detaljnijeNovosti), ENT\_QUOTES, 'UTF-8');?>')" >Detaljnije...</a>*

*<?php } ?>*

Funkcija PrikaziPunuNovost šalje ajax zahtjev post metodom sa parametrima php skripti punaNovost.php koja prikazuje samo novost koja je kliknuta sa punim tekstom:

*function PrikaziPunuNovost(datum,naslov,autor,slika,sadrzajNovosti,detaljnijeNovosti)*

*{ var ajax = new XMLHttpRequest();*

*ajax.onreadystatechange = function () {*

*if (ajax.readyState == 4 && ajax.status == 200) {*

*document.body.innerHTML = ajax.responseText;*

*}*

*if (ajax.readyState == 4 && ajax.status == 404)*

*document.body.innerHTML = "Greska: nepoznat URL";*

*}*

*slika=encodeURIComponent(slika);*

*ajax.open("POST","punaNovost.php", true);*

*ajax.setRequestHeader("Content-type","application/x-www-form-urlencoded");*

*ajax.send("datum="+datum+"&naslov="+naslov+"&autor="+autor+"&slika="+slika+"&sadrzajNovosti="+sadrzajNovosti+"&detaljnijeNovosti="+detaljnijeNovosti);*

*return false;*

*}*

Tako je omogućeno da kada se klikne na link Detaljnije da se prikaže detaljnije samo ta novost asinhrono bez refresha stranice.

Skripta punaNovost.php sadrži sve što ostale stranice samo što u sekciji glavni\_dio sadrži novost koja je primljena post metodom tako da je zbog prostora nećemo ovdje navoditi.

Nakon što se ispitalo u datoteci koja prikazuje Novosti ima li detaljnijeg teksta ispituje se ima li linka sa slikom ako ima prikazuje se:

*<?php if(trim($slika)!="") { ?>*

*<img class="novosti\_slika" src=<?php echo '"'.htmlspecialchars(trim($slika), ENT\_QUOTES, 'UTF-8').'"'; ?> width="500" height="300" alt="slika"> <?php } ?>*

I ostalo je još da se prikaže autor, datum i vrijeme novosti što je urađeno sljedećim kodom čime se i završava skripta prikaziIzDat.php koju smo u potpunosti naveli i objasnili :

*<p class="datum"><?php echo htmlspecialchars(trim($autor), ENT\_QUOTES, 'UTF-8').' , '. htmlspecialchars(trim($datum), ENT\_QUOTES, 'UTF-8') ?> </p>*

*</div>*

*<?php } ?>*

5. Peta spirala

Novosti u bazi

U ovom spirali je implementirano to da novosti umjesto u txt datoteka držimo u MySQL bazi podataka. Zbog toga sam kreirao u phpMyAdmin bazu podataka (kao što je objašnjeno na tutorijalu br. 8 tako da postupak neću ovdje navoditi) sa nazivom novosti koja sadrži tabelu novost sa atributima koji je opisuju: id, naslov, tekst, autor, vrijeme, link slike i detaljan tekst. Sada u datoteci Naslovna.php umjesto uključivanja skripte koja prikazuje novosti iz txt datoteka uključena je skrpita NovostiIzBaze.php, koja prikazuje novosti iz baze podataka koja će biti objašnjena u nastavku, a NovostiIzDat.php je sada zakomentarisana.

*<div id="novosti">*

*<?php //include "NovostiIzDat.php"; ?>*

*<?php include "NovostiIzBaze.php"; ?>*

*</div>*

Sadržaj skripte NovostiIzBaze.php je sljedeći. Prvo se spajamo na bazu koju smo kreirali, tako što kreiramo PDO objekat na sljedeći način:

*<?php*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

Među parametrima ovog objekta su adresa MySQL servera na koji se konektujemo, a pošto je u pitanju isti računar koristimo localhost. Baza koju smo kreirali, a to je novosti, charset, odnosno kodiranje slova je postavljeno na utf8 kako bi radila naša slova. Ovdje su navedeni i korisničko ime i šifra za konekciju na bazu.

Kreirani PDO objekat posjeduje funkciju query koja služi za izvršavanje SQL upita i koja vraća rezultat:

*$rezultat = $veza->query("select id, naslov, tekst, UNIX\_TIMESTAMP(vrijeme) vrijeme2, autor, slika, detaljantekst from novost order by vrijeme desc");*

*if (!$rezultat) {*

*$greska = $veza->errorInfo();*

*print "SQL greška: " . $greska[2];*

*exit();*

*}*

sada nam je u varijabli $rezultat vraćeni novosti iz tabele sortirani po vremenu objavljivanja što smo postigli sa „from novost order by vrijeme desc“ djelom upita.

U nastavku prolazimo foreach petljom kroz dobivene novosti i prikazujemo ih kao i prije:

*foreach ($rezultat as $novost) { ?>*

*<div class="novost">*

*<h1 class="naslov"><?php echo $novost['naslov']; ?></h1>*

*<p class="tekst\_novosti"> <?php echo $novost['tekst']; ?></p>*

*<?php if($novost['detaljantekst']!="") { ?>*

*<a onclick="return PrikaziPunuNovost('<?php echo date("d.m.Y. (h:i)", $novost['vrijeme2']);?>','<?php echo $novost['naslov'];?>','<?php echo $novost['autor'];?>','<?php echo $novost['slika'];?>','<?php echo $novost['tekst'];?>','<?php echo $novost['detaljantekst'];?>')" >Detaljnije...</a>*

*<?php } ?>*

*<br>*

*<?php if(trim($novost['slika']!="")) { ?>*

*<img class="novosti\_slika" src=<?php echo '"'.$novost['slika'].'"'; ?> width="500" height="300" alt="slika"> <?php } ?>*

*<p class="datum"><?php echo $novost['autor'].' , '. date("d.m.Y. (h:i)", $novost['vrijeme2']); ?> </p>*

Komentari na novosti

U Bazi podataka smo kreirali novu tabelu sa nazivom komentar koja sadrži potrebne podatke i vezu na tabelu novost koja je izvedena tako što tabela komentar sadrži vijest koja je strani ključ i povezana je na id iz tabele novost. Da bismo prikazali ispod svake novosti broj komentara u nastavku je pisan sljedeći kod:

*$broj=$veza->prepare("select count(\*) from komentar where novost=?");*

*$broj->bindValue(1,$novost['id'],PDO::PARAM\_INT);*

*$broj->execute();*

*$broj=$broj->fetchColumn();*

*if ($broj==0) print "<small> Nema komentara </small>";*

*else print "<a href='komentari.php?id=".$novost['id']."'>" ."<small>".$broj." komentara </small> </a>";*

ovdje smo koristili pripremljeni upit da bi izbjegli sigurnosni propust SQL injection pa smo umjesto query koristili prepere i umjesto da navedemo where novost=$novost['id'], naveli smo novost=?, a ,$novost['id'] u bindValue djelu.

Varijabla $broj predstavlja broj komentara iz tabele koji odgovaraju jednoj novosti i prikazuju se kao link na novu skriptu komentari.php kojoj se šalje id novosti preko parametra i koja prikazuje samo komentare odabrane novosti zasebno. Sadržaj skripte komentari.php koja prikazuje komentare na vijest je sljedeći, a u kodu je sve poznato od ranijih objašnjenja:

*<?php*

*$idnovosti=$\_GET['id'];*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

*$rezultat = $veza->prepare("SELECT tekst,autor,vrijeme,email FROM komentar WHERE novost=? order by vrijeme desc");*

*$rezultat->bindValue(1,$idnovosti,PDO::PARAM\_INT);*

*$rezultat->execute();*

*if (!$rezultat) {*

*$greska = $veza->errorInfo();*

*print "SQL greška: " . $greska[2];*

*exit();*

*}*

*foreach ($rezultat->fetchAll() as $komentar) {*

*print $komentar['tekst']." , ".$komentar['autor']." , ".$komentar['vrijeme']." , ".$komentar['email']."<br>";*

*}*

*?>*

Dalje nakon što se prikazuje broj komentara ispod svake novosti je data i forma za unos komentara u nastavku skripte NovostiIzBaze.php:

*print "<br><br>";*

*print '<form id="komentar\_forma" action="Naslovna.php" method="post">';*

*print '<p>Komentar: <textarea id="tekstkomentara" name="tekstkomentara"></textarea>';*

*print ' <p>autor: &nbsp;*

*<input id="autorkomentara" name="autorkomentara" type="text"> ';*

*print ' <p>e-mail: &nbsp;*

*<input id="emailkomentara" name="emailkomentara" type="text"> ';*

*print ' <input id="novost" name="novost" type="hidden" value="'.$novost["id"].'"> ';*

*print '<input id="posalji" type="submit" value="Pošalji">';*

*print '</form>';*

*print "</div>";*

*}*

Koja sadrži skriveno polje novost koja prosljeđuje id novosti na koji ćemo dodati komentar. I kada se pritisne submit button Pošalji od forme za dodavanje komentara aktivira se sljedeći kod koji je u nastavku:

*if (isset($\_REQUEST['tekstkomentara'])) {*

*$novi\_komentar=$\_REQUEST['tekstkomentara'];*

*$novi\_autor=$\_REQUEST['autorkomentara'];*

*$novi\_email=$\_REQUEST['emailkomentara'];*

*$veza->exec("INSERT INTO komentar SET novost=".$\_POST['novost'].",tekst='$novi\_komentar', autor='$novi\_autor', email='$novi\_email'");*

*}*

*?>*

koji dodaje novi komentar u tabelu komandom INSERT INTO komentar sa novošću čiji je id prosljeđen iz skrivenog polja forme za unos komentara post metodom.

Admin panel

Na naslovnoj strani u Naslovna.php smo dodali dio za prijavljivanje administratora sa korisničkim imenom i šifrom. To je urađeno tako što je dodan ovaj kod ispid zaglavlja:

<div id="LogInDio">

<form class="LogIn" method="post" action="login.php">

<input type="text" placeholder="Korisničko ime:" name="korisnickoime">

<input type="text" placeholder="Šifra:" name="sifra">

<input type="submit" value="Log in" name="submitLogin">

</form>

<br>

</div>

A u Stil.css datoteci su dodane osobine koje smještaju dio za prijavu na desnu stranu stranice:

*.LogIn {*

*position:absolute;*

*right:15px;*

*}*

Pritiskom na submit dugme za prijavu pokreće se login.php skripta kojoj se prosljeđuju uneseno korisničko ime i unesena šifra. U našoj bazi podataka smo kreirali tabelu korisnici koja sadrži polja korisnička šifra koja je primarni ključ jer se ne smije ponavljati i polje šifra tipa varchar. Za prijavu koristimo sesije pa je na početku login.php skripte navedeno:

*<?php*

*session\_start();*

Zatim provjeravamo da li je administrator već logiran odnosno da li $\_SESSION['login\_user'] ima neku vrijednost:

*if (isset($\_SESSION['login\_user'])) {*

*$korisnik= $\_SESSION['login\_user'];*

*header("location: Admin.php");*

*}*

Ako ima tj. ako je administrator logiran upisujemo u varijablu $korisnik i otvaramo Admin panel, tj. datoteku pod imenom Admin.php koja će biti objašnjena u nastavku. Ako administrator nije logiran provjeravamo je li uneseno korisničko ime i šifra i da li se podudara sa podacima iz baze podataka.

*else*

*{*

*if(empty($\_POST['korisnickoime']) || empty($\_POST['sifra']))*

*{*

*echo "Neispravno korisnicko ime ili sifra";*

*}*

*else*

*{*

*$korisnik=$\_POST['korisnickoime'];*

*$sifra= $\_POST['sifra'];*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

*$rezultat = $veza->prepare("SELECT \* FROM korisnici WHERE username=? and password=?");*

*$rezultat->execute(array($korisnik,$sifra));*

*$rezultat=$rezultat->fetchColumn();*

Spojili smo se na bazu i kreirali pripremljeni upit da bi smo izbjegli SQL injection da bismo dobili rezultat koji odgovara podacima koji odgovaraju unesenimu varijablu *$rezultat* . Ako takav rezultat postoji tj. ako je ($rezultat!="") postavljamo korisnilko ime u sesiju i otvaramo admin panel. Na kraju ako nema takvog rezultata u tabeli ispisuje se da je neispravno korisničko ime ili šifra.

*if ($rezultat!="") {*

*$\_SESSION['login\_user']=$korisnik;*

*header("location: Admin.php");*

*}*

*else {*

*print "Neispravno korisnicko ime ili sifra";*

*}*

*}*

*}*

*?>*

Sada ćemo objasniti admin panel toj datoteku admin.php.

Pored onoga što sadrže ostale stranice(zaglavlje, meni i sl.) admin.php sadrži dio koji ispisuje korisničko ime administratora i dio za odjavu:

*<div id="LogInDio">*

*<p> Prijavljeni ste kao: <?php session\_start(); print $\_SESSION['login\_user']?> <br>*

*<form class="LogIn" method="post" action="admin.php">*

*<input type="submit" value="Log out" name="submitLogout">*

*</form>*

*<?php if (isset($\_POST['submitLogout'])) {*

*session\_destroy();*

*header("Location: Naslovna.php");*

*}*

*?>*

*<br>*

*</div>*

Sa *if (isset($\_POST['submitLogout']))* provjeravamo da li je pritisnuto dugme za odjavu, ako jeste zatvaramo sesiju i vraćamo se na našu Naslovnu.php. Dalje u sekciji glavni\_dio admin.php sadrži formu za unos novosti koja ima tri buttona za submit: Dodaj, Izmjeni i Obriši koja omogućava korisniku da unosi novu novost, da mjenja i briše postojeću na osnovu id-a. Forma izgleda ovako:

*<div id="glavni\_dio">*

*<h2> Unesite novost: </h2>*

*<form id="forma\_novosti" action="Admin.php" method="post">*

*<p>Naslov: &nbsp*

*<input id="naslovnovosti" name="naslovnovosti" type="text">*

*<p>Tekst: <textarea id="tekstnovosti" name="tekstnovosti"></textarea>*

*<p>Detalajn tekst: <textarea id="detaljantekstnovosti" name="detaljantekstnovosti"></textarea>*

*<p>Autor: &nbsp*

*<input id="autornovosti" name="autornovosti" type="text">*

*<p>URL Slike: &nbsp*

*<input id="slikanovosti" name="slikanovosti" type="text">*

*<p>ID novosti koju želite obrisati ili izmjeniti: &nbsp*

*<input id="idnovosti" name="idnovosti" type="input"> <br>*

*<input name="dodaj" type="submit" value="Dodaj">*

*<input name="izmjeni" type="submit" value="Izmjeni">*

*<input name="obrisi" type="submit" value="Obriši">*

*</form>*

a dio koda koji osluškuje buttone i radi navedene operacije je sljedeći:

*<?php //dodavanje,brisanje i izmjena novosti*

*if(isset($\_POST['dodaj'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

*$veza->exec("INSERT INTO novost SET naslov='".$\_POST['naslovnovosti']."',tekst='".$\_POST['tekstnovosti']."',autor='".$\_POST['autornovosti']."',slika='".$\_POST['slikanovosti']."',detaljantekst='".$\_POST['detaljantekstnovosti']."'");*

*}*

*if(isset($\_POST['obrisi'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("delete from novost where id='".$\_POST['idnovosti']."'");*

*$veza->exec("delete from komentar where novost='".$\_POST['idnovosti']."'");*

*}*

*if(isset($\_POST['izmjeni'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$ps=$veza->prepare("UPDATE novost SET naslov='".$\_POST['naslovnovosti']."',tekst='".$\_POST['tekstnovosti']."',autor='".$\_POST['autornovosti']."',slika='".$\_POST['slikanovosti']."',detaljantekst='".$\_POST['detaljantekstnovosti']."'"."WHERE id='".$\_POST['idnovosti']."'");*

*$ps->execute();*

*}*

U ovom djelu koda kada obrišemo neku novost brišemo i komentare na tu novost iz baze naredbom delete from.

Dalje što je urađeno na admin panelu je to da su navedeni svi komentari sa dugmatima za brisanje pored svakog komentara koji se zove „Obriši komentar“. Da bi smo prikazali sve komentare spojili smo se na bazu i uzeli sve komentare iz baze poredane po vremenu objavljivanja u varijablu $rezultat:

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

*$rezultat = $veza->query("SELECT novost,tekst,autor,vrijeme,id FROM komentar order by vrijeme desc");*

Nakon toka prolazili smo foreach petljom kroz komentare i ispisivali svaki pojedinačno i uz svakog button za njegovo brisanje gdje se nalazi i skriveni input preko kojeg se prenosi id komentara kojeg ćemo brisati:

*print "<h2>Komentari:</h2>";*

*foreach ($rezultat->fetchAll() as $komentar) {*

*print $komentar['tekst']." , ".$komentar['autor']." , ".$komentar['vrijeme']."<form id='forma\_komentara' action='Admin.php' method='post'> <input name='obrisikomentar' type='submit' value='Obriši komentar'> <input name='idkomentara' type='hidden' value='".$komentar['id']."'></form> <br>";*

*}*

Nakon toga provjeravamo da li je pritisnuto dugme za brisanje i ako jeste brišemo komentar koji se nalazio uz pritisnuto dugme:

*if(isset($\_POST['obrisikomentar'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("delete from komentar where id='".$\_POST['idkomentara']."'");*

*}*

Na analogan način kao i za novosti prikazana je forma za Dodavanje, Brisanje i Izmjenu Korisnika koja radi na istom principu kao za novosti koja je već objašnjena:

*//dodavanje,promjena brisanje korisnika*

*?>*

*<h2>Unesite korisnika:</h2>*

*<form class="Korisnici" method="post" action="admin.php">*

*<input type="text" placeholder="Korisničko ime:" name="korisnickoime">*

*<input type="text" placeholder="Šifra:" name="sifra"><br>*

*<input type="submit" name="dodajkorisnika" value="Dodaj">*

*<input type="submit" name="izmjenikorisnika" value="Izmjeni">*

*<input type="submit" name="obrisikorisnika" value="Obriši">*

*</form>*

*<?php*

*if(isset($\_POST['dodajkorisnika'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("set names utf8");*

*$veza->exec("INSERT INTO korisnici SET username='".$\_POST['korisnickoime']."',password='".$\_POST['sifra']."'");*

*}*

*if(isset($\_POST['izmjenikorisnika'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$ps=$veza->prepare("UPDATE korisnici SET username='".$\_POST['korisnickoime']."',password='".$\_POST['sifra']."'");*

*$ps->execute();*

*}*

*if(isset($\_POST['obrisikorisnika'])) {*

*$veza = new PDO("mysql:dbname=novosti;host=localhost;charset=utf8", "NovostiUser", "novostipass");*

*$veza->exec("delete from korisnici where username='".$\_POST['korisnickoime']."'");*

*}*

*?>*

6. šesta spirala