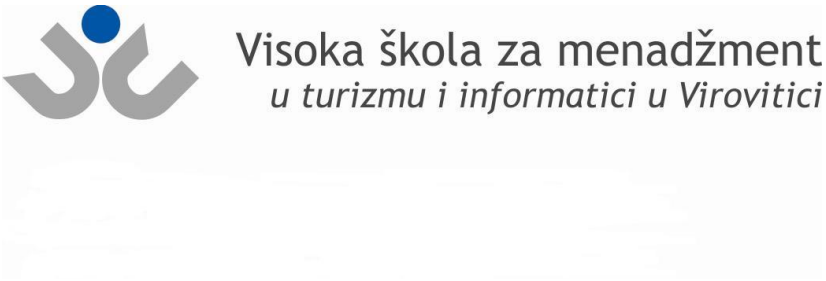


PRIRUČNIK ZA LABORATORIJSKE VJEŽBE IZ PREDMETA

OSNOVE WEB PROGRAMIRANJA

5. laboratorijska vježba

dr.sc. Oliver Jukić, prof.v.š
Ivan Heđi, dipl.ing., pred.
Ivan Špeh, mag.ing.el.
Antonio Šarabok, mag.ing.comp.

Virovitica, prosinac 2017.

Urednik
dr. sc. Oliver Jukić, prof.v.š.

Materijal nije recenziran i služi samo za internu uporabu

Grafička priprema
Ivan Heđi, dipl.ing., pred.
Ivan Špeh, mag.ing.el.
Antonio Šarabok, mag.ing.comp.

OSNOVNE UPUTE

- Laboratorijske vježbe (LV) održavat će se kroz 4.-8. tjedna nastave, u trajanju od 15 sati, dinamikom od 3 sata tjedno.

- Tijekom LV-a student će se u praksi upoznavati s pojedinim elementima HTML-a, CSS-a i Javascripta, koje će tijekom konstrukcijskih vježbi integrirati u cjeloviti projekt.

- Laboratorijske vježbe su iste za sve studente i izvode se prema zadanim pripremama i uputama za rad u laboratoriju.

- Zabranjeno je korištenje Interneta na računalu (osim za potrebe vježbe kao npr. Google, Office365, Stackoverflow i sl.) i korištenje mobitela. Nepoštivanje ove odredbe povlači udaljavanje s LV.

- Student ima obvezu kolokvirati svaku laboratorijsku vježbu, što je uvjet za potpis u indeks kao i za obavljanje konstrukcijskih vježbi.

- Student je dužan popuniti i proučiti pripremu za pojedinu laboratorijsku vježbu, na temelju koje će pisati kratku provjeru znanja (blic) u trajanju od 10 minuta na sustavu za e-učenje. Rezultati bliceva biti će objavljeni odmah.

- U slučaju nedovoljnog broja bodova na blicu student gubi pravo odrađivanja laboratorijske vježbe, ali će imati priliku ponovljenog blica i obavljanja vježbe u terminu kojeg odredi profesor.

- Nakon blica, student odrađuje laboratorijsku vježbu, koja se sastoji od obveznih i dodatnih zadataka. U pripremi će biti točno naznačeno koje zadatke student mora riješiti kako bi vježba bila uspješno kolokvirana. Kolokviranje podrazumijeva razumijevanje svake linije generiranog programskog koda.

- Ukoliko student ne riješi sve obvezne zadatke tijekom laboratorijske vježbe, mora ih riješiti doma te ih naknadno (u pravilu sutradan) prezentirati nastavniku u unaprijed objavljenom terminu.

- Ukoliko student ne riješi sve dodatne zadatke tijekom laboratorijske vježbe, mora ih riješiti doma te ih prezentirati nastavniku u terminu sljedećih laboratorijskih vježbi.

4. Laboratorijska vježba

Sadržaj vježbe:

- AJAX općenito
- Sinkrono i asinkrono izvršavanje koda
- jQuery AJAX

AJAX

Mnoge moderne web stranice i gotovo sve web aplikacije koriste tehnologiju zvanu AJAX (eng. Asynchronous JavaScript and XML) kako bi u pozadini ”dohvatile” podatke sa poslužitelja i na dinamičan način ažurirale sadržaj na stranici. Podaci koji se dohvaćaju sa AJAX-om mogu biti gotovi HTML elementi, XML, json ili bilo koji drugi tekstualni zapis podataka. AJAX tehnika se razlikuje od konvencionalnog web programiranja na slijedeći način:

- **Konvencionalna web stranica** sadrži veze ili obrasce koji, kada se kliknu ili šalju, upućuju zahtjev novom URL-u na web poslužitelju. Poslužitelj tada odgovara slanjem cijele nove HTML stranice, koju preglednik zatim prikazuje, zamjenjujući originalnu stranicu. To može dugo trajati i ometati korisniku, jer korisnik mora čekati dok se nova stranica učitava.
- Uz pomoć **AJAX tehnike** se također postavlja zahtjev URLu na poslužitelju uz razliku što se može mjenjati dio sadržaja stranice, bez ponovnog učitavanja. Tako se korisnicima može npr. prikazati obavijest o učitavanju podataka sa poslužitelja dok se u pozadini obavlja komunikacija između klijenta i servera ili radi obrada podataka učitanih sa poslužitelja. Tako se korisnik može služiti web stranicom ili aplikacijom dok se vrši učitavanje podataka, što kod konvencionalnog pristupa nije slučaj. Ovakav pristup korisnicima pruža neometan rad u aplikaciji.

Sinkrono i asinkrono izvršavanje koda

Uporaba AJAX-a na stranici donosi neka ograničenja koja programer treba imati na umu, a najbitnije od svih je asinkrono izvršavanje koda. Da bismo razumjeli asinkrono ponašanje koda pogledajmo primjer sinkronoga koda:

```
console.log('Baze podataka');
console.log('Osnove tehničkih sustava');
console.log('Programiranje u .NET okolini');
console.log('Osnove web programiranja');
```

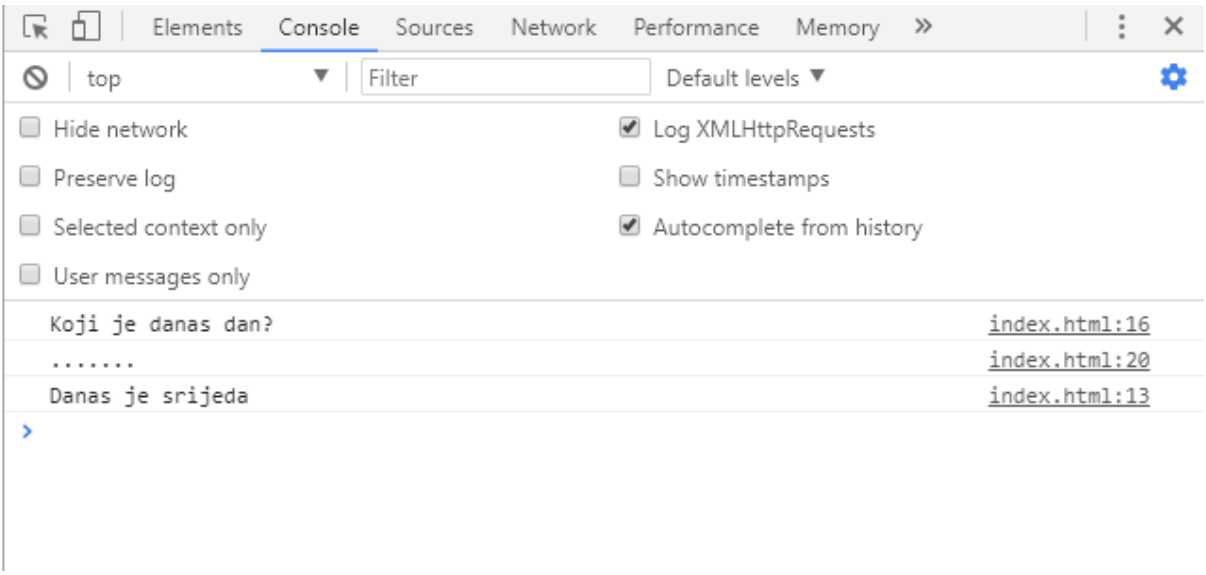
U prikazanome primjeru kod će se izvršavati po redu kako je napisan – od ispisa ”Baze podataka” do ispisa ”Osnove web programiranja”. U slijedećem primjeru je prikazan primjer sa asinkronim izvršavanjem koda:

```
console.log('Koji je danas dan?');

setTimeout(function () {
      console.log('Danas je srijeda');
}, 200);

console.log('.......');
```

U prvoj liniji koda ispisuje se tekst "Koji je danas dan?" u konzolu. Zatim je pozvana funkcija `setTimeout()` koja kao prvi parametar prima funkciju koja će se izvršiti nakon vremena postavljenog kao drugi parametar – 200ms. Nakon poziva `setTimeout()` funkcije poziva se ispis teksta (točkice). Pokrenemo li ovaj kod, u konzolu se ispisuje slijedeći sadržaj:



Slika 1. Rezultat izvršavanja asinkronog koda

Bez obzira što se funkcija za ispis točkica nalazi ispod poziva `setTimeout()` funkcije, točkice su se ispisale prije ispisa dana, odnosno nisu čekale izvršavanje `setTimeout()` funkcije. Vrijeme čekanja je u razvoju web-a vrlo važno vrijeme koje programer može iskoristiti za prikaz obavijesti, uputa, upozorenja i sl.

jQuery AJAX

Čisti Javascript AJAX kod je u većini slučajeva "nečitljiv", stoga ćemo koristiti jQuery-evu `$.ajax()` funkciju za AJAX funkcionalnosti, čiji je opći oblik prikazan u nastavku na slici 2.

```
$.ajax({
        url: 'http://localhost/LV5/action.php',
        type: 'POST',
        dataType: 'html',
        data:
        {
                akcija: 'naziv_akcije',
                varijabla_1: 500.23,
                varijabla_2: 'vrijednost varijable 2',
                varijabla_3: 'vrijednost varijable 3',
        },
        success: function (sOdgovorPosluzitelja)
        {
                //asinkroni poziv ukoliko je došlo do odgovora sa servera
                Funkcija_1();
                Funkcija_2();
        },
        error: function(XMLHttpRequest, textStatus, exception)
        {
                //asinkroni poziv ukoliko nije došlo do odgovora sa servera
                console.log('Došlo je do pogreške');
        },
});
```

Slika 2. Opći oblik jQuery AJAX-a

Osnovne AJAX funkcionalnosti u jQuery-u su opisane sa 6 atributa pomoću kojih govorimo pregledniku što želimo napraviti:

- `url` – Lokacija prema kojoj radimo zahtjev.
- `type` – Tip http zahtjeva koji pravimo prema poslužitelju. Postoji više tipova zahtjeva, a osnovni su:
  - `POST` – služi za slanje podataka (koristi se kada se želi spremiti novi podatak)
  - `GET` – služi za dohvaćanje podataka (koristi se kada se želi dohvatiti podatke)
  - `PUT` – služi za ažuriranje postojećih podaka
  - `DELETE` – služi za brisanje podataka
- `dataType` – Tip podatka koji se očekuje kao odgovor od poslužitelja. Može biti html, json, xml i sl.
- `data` – podaci u obliku Javascript objekta koje šaljemo prema poslužitelju
- `success` – asinkrona funkcija koja se poziva nakon što se dobije poruka od poslužitelja
- `error` – asinkrona funkcija koja se poziva ukoliko dođe do pogreške prilikom komunikacije sa poslužiteljom. Najčešće dolazi uslijed krivo napisanog url-a poslužitelja.

Na temelju prikazanog općeg oblika jQuery AJAX-a dohvatit ćemo listu studenata sa poslužitelja te ćemo na istu dodati nove studente, pri čemu ćemo ih prikazati u HTML tablici. Na poslužitelju se nalaze datoteke `studenti.json` i `action.php`. Uz pomoć skripte `action.php` ćemo dohvaćati i dodavati studente, dok će nam datoteka studenti.json služiti kao ”baza podataka”.

Poslužiteljska strana

studenti.json

```
[
  {
    "oib": "12345",
    "ime": "Ivan",
    "prezime": "Peric",
    "godina_studija": "1",
    "smjer": "racunarstvo"
  },
  {
    "oib": "12346",
    "ime": "Josip",
    "prezime": "Horvat",
    "godina_studija": "2",
    "smjer": "menadzment"
  },
  {
    "oib": "118877",
    "ime": "Petar",
    "prezime": "Kovac",
    "godina_studija": "3",
    "smjer": "poduzetnistvo"
  }
]
```

action.php

```php
<?php

$sAkcija = $_POST['akcija'];

switch($sAkcija)
{
	case 'dodaj_studenta':
		//dohvaćanje podataka iz forme
		$sOIB = $_POST['oib'];
		$sIme = $_POST['ime'];
		$sPrezime = $_POST['prezime'];
		$sGodinaStudija =$_POST['godina_studija'];
		$sSmjer = $_POST['smjer'];

		//dodvaćanje json datoteke sa studentima i pretvorba string -> polje
		$sJsonFile = file_get_contents('./studenti.json', FILE_USE_INCLUDE_PATH);
		$oJson = json_decode($sJsonFile);

		//kreiranje novog člana polja sa dohvaćenim podacima
		$oSentData = array
		(
			"oib"			=>	$sOIB,
			"ime"			=>	$sIme,
			"prezime"		=>	$sPrezime,
			"godina_studija"	=>	$sGodinaStudija,
			"smjer"			=>	$sSmjer
		);
		array_push($oJson, $oSentData);

		//vraćanje polja u json string i zapis u json datoteku
		$sJsonString = json_encode($oJson);
		file_put_contents('./studenti.json', $sJsonString);

		//odgovor poslužitelja (upisani podaci)
		echo 'Uspješno ste dodali studenta: ';
		echo ' OIB: '.$sOIB;
		echo ' Ime: '.$sIme;
		echo ' Prezime: '.$sPrezime;
		echo ' Godina studija: '.$sGodinaStudija;
		echo ' Smjer: '.$sSmjer;
	break;

	case 'dohvati_tablicu_studenata':
		//dodvaćanje json datoteke sa studentima i pretvorba string -> polje
		$sJsonFile = file_get_contents('./studenti.json', FILE_USE_INCLUDE_PATH);
		$oJson = json_decode($sJsonFile);

		//odgovor poslužitelja (html tablica)
		echo
		'<table class="table table-striped">
		    <thead>
		      <tr>
		        <th>OIB</th>
		        <th>Ime</th>
		        <th>Prezime</th>
		        <th>Godina</th>
		        <th>Smjer</th>
		      </tr>
		    </thead>
		    <tbody>';

		foreach($oJson as $oStudent)
		{
```

```
                    echo
                    '<tr>
                        <td>'.$oStudent->oib.'</td>
                        <td>'.$oStudent->ime.'</td>
                        <td>'.$oStudent->prezime.'</td>
                        <td>'.$oStudent->godina_studija.'</td>
                        <td>'.$oStudent->smjer.'</td>
                     </tr>';
                }

                echo
                '</tbody>
                </table>';
        break;
}

?>
```

Klijentska strana

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>AJAX</title>
    <script src="assets/plugins/jquery/jquery-3.2.1.min.js"></script>
    <link rel="stylesheet" href="assets/plugins/bootstrap-3.3.7-dist/css/bootstrap.css">
</head>
<body>

        <div class="container" style="width: 30%">

                <h2>Dodaj studenta</h2>

                <div class="form-group">
                        <label for="inptOIB">OIB</label>
                        <input type="number" class="form-control" id="inptOIB"
placeholder="Unesite OIB">
                </div>

                <div class="form-group">
                        <label for="inptIme">Ime</label>
                        <input type="text" class="form-control" id="inptIme"
placeholder="Unesite ime">
                </div>

                <div class="form-group">
                        <label for="inptPrezime">Prezime</label>
                        <input type="text" class="form-control" id="inptPrezime"
placeholder="Unesite prezime">
                </div>

                <div class="form-group">
                        <label for="rbGodina">Godina studija</label><br/>
                        <label>
                                <input type="radio" name="rbGodina" value="1" checked> 1.
godina
                        </label><br/>
                        <label>
                                <input type="radio" name="rbGodina" value="2"> 2. godina
```

```
                        </label><br/>
                        <label>
                                <input type="radio" name="rbGodina" value="3"> 3. godina
                        </label>
                </div>

                <div class="form-group">
                        <label for="rbSmjer">Smjer</label><br/>
                        <label>
                                <input type="radio" name="rbSmjer" value="racunarstvo"
checked> Računarstvo
                        </label><br/>
                        <label>
                                <input type="radio" name="rbSmjer" value="menadzment">
Menadžment
                        </label><br/>
                        <label>
                                <input type="radio" name="rbSmjer" value="poduzetnistvo">
Poduzetništvo
                        </label>
                </div>


                <button class="btn btn-success" onclick="SpremiStudenta()">Pošalji</button>

                <div id="obavijest" style="margin-top:15px;"></div>

                <h2>Pregled studenata</h2>
                <div id="tablica" style="margin-top:15px;"></div>
        </div>


        <script src="assets/plugins/bootstrap-3.3.7-dist/js/bootstrap.min.js"></script>
        <script>

        function UcitajTablicu()
        {
                var oTablica = $('#tablica');
                oTablica.empty();
                $.ajax({
                        url: 'http://localhost/LV5/action.php',
                        type: 'POST',
                        dataType: 'html',
                        data:
                        {
                                akcija: 'dohvati_tablicu_studenata'
                        },
                        success: function (sOdgovorPosluzitelja)
                        {
                                oTablica.html(sOdgovorPosluzitelja);
                        },
                        error: function(XMLHttpRequest, textStatus, exception)
                        {
                                console.log('Došlo je do pogreške pri učitavanju tablice');
                        },
                        async: true

                });
        }

        function SpremiStudenta()
        {
                var sOIB = $('#inptOIB').val();
                var sIme = $('#inptIme').val();
                var sPrezime = $('#inptPrezime').val();
                var sGodinaStudija = $('input[name=rbGodina]:checked').val();
```

```
                var sSmjer = $('input[name=rbSmjer]:checked').val();

                if(sOIB == '' || sIme == '' || sPrezime == '')
                {
                        alert('Morate popuniti sva polja');
                        return false;
                }
                else
                {
                        $.ajax({
                                url: 'http://localhost/LV5/action.php',
                                type: 'POST',
                                dataType: 'html',
                                data:
                                {
                                        akcija: 'dodaj_studenta',
                                        oib: sOIB,
                                        ime: sIme,
                                        prezime: sPrezime,
                                        godina_studija: sGodinaStudija,
                                        smjer: sSmjer
                                },
                                success: function (sOdgovorPosluzitelja)
                                {
                                        $('#obavijest').html('<div class="alert alert-success
alert-dismissable fade in"> <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a> Student uspješno dodan  </div>');
                                        console.log(sOdgovorPosluzitelja);
                                        UcitajTablicu();
                                },
                                error: function(XMLHttpRequest, textStatus, exception)
                                {
                                        $('#obavijest').html('<div class="alert alert-danger
alert-dismissable fade in"> <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a> Došlo je do pogreške</div>');
                                },
                                async: true

                        });
                }
        }
        $(document).ready(function(){
                UcitajTablicu();
        });
    </script>
</body>
</html>
```

U datoteci `index.html` je dodana Bootstrap forma i dvije Javascript funkcije:

- `UcitajTablicu()` - prikazuje tablicu sa studentima.
- `SpremiStudenta()` - sprema upisane podatke studenta iz forme u datoteku `studenti.json` na poslužitelju.

Funkcija `UcitajTablicu()` se poziva odmah nakon što se učita stranica i prikazuje tablicu sa studentima ispod Bootstrap forme. Nakon što korisnik unese podatke studenta u formu i klikne na gumb "Pošalji" poziva se funkcija `SpremiStudenta()`. Nakon uspješne validacije, šalju se podaci iz forme prema poslužitelju koji ih sprema u bazu podataka `studenti.json`. Nakon uspješne pohrane podataka, šalje se poruka prema poslužitelju koja se ispisuje u konzolu, unutar `success` funkcije. Budući da je dodan novi student, potrebno je osvježiti tablicu pozivom funkcije `UcitajTablicu()`.

Priprema za vježbu

1. Što je AJAX i koji problem rješava?

2. Objasnite razliku između sinkronog i asinkronog izvršavanja koda.

3. Napišite Javascript kod koji će svake sekunde u konzolu ispisati parni broj. Krenuti od broja 0, stati na broju 50. Koristiti `setTimeout()` funkciju unutar for petlje. Što se ispisalo u konzolu?

4. Kojim redosljedom će se ispisati tekst generiran slijedećim kodom? Objasnite!

```
console.log('Nedjelja');

setTimeout(function () {
       console.log('Ponedjeljak');
}, 0);

console.log('Subota');
```

5. Napišite opći oblik jQuery sintakse.

6. Napišite opći oblik jQuery AJAX funkcije.

7. Navedite i opišite osnove atribute koje koristimo unutar jQuery AJAX funkcije.

8. Navedite tipove http zahtjeva koji se koriste unutar `type` atributa.

9. Napišite AJAX kod koji će u konzolu ispisati sadržaj sa URL-a
   `https://jsonplaceholder.typicode.com/posts`

10. Nadopunite funkciju `success` u prethodnom zadatku na način da naziv svakoga posta prikažete na stranici u zasebnom `<p>` elementu

```
success: function (sOdgovorPosluzitelja)
{
```

```
}
```

Zadaci za rad u laboratoriju

Prije početka rješavanja zadataka potrebno je preuzeti prilog vezan za vježbu u kojem se nalazi predložak sa strukturom datoteka i mapa. Kreirati mapu LV5 unutar korisničke mape te raspakirati preuzeti prilog u novokreiranu mapu.

Obavezni zadaci

Zadatak 1. Prikaz vijesti na naslovnici

U datoteci `vijesti.js` potrebno je napisati funkciju `UcitajVijestiNaslovnice()` koja će u sebi sadržavati AJAX kod za dohvaćanje podataka sa URL-a

`http://localhost/{ImePrezime}/LV5/json.php`

Unutar `success` metode potrebno je pretvoriti odgovor sa poslužitelja u polje Javascript objektata koristeći jQuery metodu `$.parseJSON()` te prikazati podatke pojedine vijesti unutar elementa `<div id="sve-vijesti">` prolaskom petlje kroz pojedini objekt. Unutar datoteke `index.html` u funkciji `$(document).ready()` pozvati funkciju `UcitajVijestiNaslovnice()` koja će svaki puta kada se stranica učita prikazati vijesti preuzete sa poslužitelja.

Zadatak 2. Administracija vijesti - HTML

Kreirati novu datoteku sa nazivom `administracija.html` te slijedeći sadržaj:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Administracija</title>
    <script src="assets/plugins/jquery/jquery-3.2.1.min.js"></script>
    <link rel="stylesheet" href="assets/plugins/bootstrap-3.3.7-dist/css/bootstrap.css">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

       <div class="container-fluid">
              <header>
                  <nav class="navbar navbar-default navbar-fixed-top">
                      <div class="container-fluid">
                          <div class="navbar-header">
                              <button type="button" class="navbar-toggle" data-
toggle="collapse"
                              data-target="#main-navbar">
                                  <span class="icon-bar"></span>
                                  <span class="icon-bar"></span>
```

```
                                <span class="icon-bar"></span>
                            </button>
                                <a class="navbar-brand" href="index.html"><img
src="img/vsmti_logo.png"></a>
                            </div>
                            <div class="collapse navbar-collapse" id="main-navbar">
                                <ul class="nav navbar-nav">
                                    <li><a href="index.html">Pregled vijesti</a></li>
                                    <li class="active"><a href="administracija.html"> <span
class="glyphicon glyphicon-edit" aria-hidden="true"></span> Administracija</a></li>
                                </ul>
                            </div>
                        </div>
                    </nav>
                </header>
<div class="administracija-container">
        <h3 class="page-header">Administracija obavijesti</h3>

        <div id="tablica-container">
                <h4>Obavijesti <button class="btn btn-sm btn-primary" data-toggle="modal"
data-target="#dodaj-vijest"><span class="glyphicon glyphicon-plus"></span></button></h4>
                <table id="tablica-postovi" class="table table-striped">
                        <thead>
                                <tr>
                                        <th>R.br.</th>
                                        <th>Datum objave</th>
                                        <th>Naziv</th>
                                        <th>Obriši</th>
                                </tr>
                        </thead>
                        <tbody>

                        </tbody>
                </table>
        </div>
</div>

        </div>

        <div class="modal fade" id="dodaj-vijest" role="dialog">
                <div class="modal-dialog">
                        <div class="modal-content">
                                <div class="modal-header">
                                        <button type="button" class="close" data-
dismiss="modal">&times;</button>
                                        <h4 class="modal-title">Dodaj vijest</h4>
                                </div>
                                <div class="modal-body">
                                        <div class="form-group">
                                                <label for="inptNazivVijesti">Naziv
vijesti</label>
                                                <input type="text" class="form-control"
id="inptNazivVijesti">
                                        </div>
                                        <div class="form-group">
                                                <label for="txtTekstVijesti">Tekst
vijesti</label>
                                                <textarea class="form-control" rows="5"
id="txtTekstVijesti"></textarea>
                                        </div>
                                </div>
                                <div class="modal-footer">
                                        <button onclick="DodajVijest()" type="button"
class="btn btn-success" data-dismiss="modal">Dodaj</button>
                                        <button type="button" class="btn btn-danger" data-
dismiss="modal">Odustani</button>
```

```
                        </div>
                    </div>
                </div>
            </div>


            <script src="assets/plugins/bootstrap-3.3.7-dist/js/bootstrap.min.js"></script>
            <script src="js/vijesti.js"></script>
            <script>
                $(document).ready(function(){
                    //UcitajTablicuAdministracija();
                });
            </script>
</body>
</html>
```

- Obrisati poziv funkcije `UcitajVijestiNaslovnice()` unutar `$(document).ready()` funkcije

Zadatak 3. Administracija vijesti – tablica sa vijestima

Nakon dodanoga HTML koda, otvoriti datoteku `vijesti.js`. Potrebno je dodati funkciju `UcitajTablicuAdministracija()` koja će u sebi sadržavati AJAX kod za dohvaćanje vijesti i slaganje istih unutar tijela tablice u stranici `administracija.html`. Unutar `success` metode potrebno je pretvoriti odgovor sa poslužitelja u polje Javascript objektata koristeći jQuery metodu `$.parseJSON()` te prikazati podatke pojedine vijesti unutar tijela tablice `<table id="tablica-postovi">`. Obratiti pozornost na ćeliju Obriši – dodati gumbić sa glyphicon-om `trash` u svaki redak na koji se klikom poziva funkcija `ObrisiVijest(nTrenutniIdVijesti)` kojom će se obrisati željena vijest iz tablice. Unutar `$(document).ready()` pozvati funkciju `UcitajTablicuAdministracija()`.

Zadatak 4. Administracija vijesti – brisanje vijesti

Unutar datoteke `vijesti.js` dodati fukciju `ObrisiVijest(nVijestID)` sa parametar id-a vijesti koji želimo obrisati. Funkcija treba sadržavati AJAX kod koji šalje zahtjev prema URLu

http://localhost/{ImePrezime}/LV5/action.php

sa podacima akcije i ID-a vijesti u obliku:

```
data:
{
        akcija: 'obrisi_vijest',
        vijest_id: nVijestID

},
```

Unutar `success` metode funkcijom alert() obavijestiti korisnika da je vijest uspješno obrisana, te pozvati funkciju `UcitajTablicuAdministracija()` kako bi se osvježila tablica sa vijestima.

Zadatak 5. Administracija vijesti – dodavanje novih vijesti

Unutar datoteke `vijesti.js` dodati fukciju `DodajVijest()` koja će dohvatiti naziv vijesti i tekst vijesti iz modala, provjeriti dali su popunjena polja, te poslati ta dva podatka sa AJAXom na url:

`http://localhost/{ImePrezime}/LV5/action.php`

Atribut `data` treba imati oblik:

```
data:
{
        akcija: 'dodaj_vijest',
        vijest_naziv: sVijestNaziv,
        vijest_tekst: sVijestTekst
},
```

Unutar `success` metode pozvati funkciju `UcitajTablicuAdministracija()`.