

HTML forme i PHP

ZADACI ZA VJEŽBU

ZADATAK 1. Napravite formu za unos podataka koja će imati proizvoljan broj polja. Polja koja moraju biti uključena su polja tipa **text**, **textarea**, **checkbox**, **radio**, **submit**, **reset**. Forma mora biti ukomponirana i uređena u HTML5 okruženju sa proizvoljnim sadržajem i jednom slikom koja će upućivati korisnika da se radi o kontakt stranici. Koristiti HTML5 deklaraciju.

ZADATAK 2. Koristite formu iz prethodnog zadatka koja će biti pozicionirana na sredini stranice horizontalno (pomoću margina), a od vrha da bude pomaknut 130px. Širina forme mora biti responzivna. Odmak unutar forme mora sa svih strana biti 10px. Radius ruba mora biti 5px.

ZADATAK 3. Korisnik kroz formu mora poslati podatke koji će se vratiti kroz obradu na serverskoj strani natrag do korisnika. Oblikovati te podatke koji se vrte na stranu korisnika tako da bude u formi stranice (zaglavlje, menu, slika, tekst).

PRIPREMA ZA LABORATORIJSKU VJEŽBU

Cilj vježbe je upoznavanje s kreiranjem HTML formi, upoznavanje s osnovama jezika PHP i kreiranjem dinamičkih web stranica. U sklopu pripreme za laboratorijsku vježbu potrebno je proučiti osnovne HTML elemente koji služe za izradu formi, te osnovnu sintaksu jezika PHP.

UPUTE ZA IZRADU LABORATORIJSKE VJEŽBE

U sklopu laboratorijske vježbe potrebno je nadograditi web stranicu iz prve laboratorijske vježbe tako da se u novom HTML dokumentu naziva **dojmovnik.html** kreira HTML formular koji će služiti za unos dojmova posjetitelja o web sjedištu ili usluzi web sjedišta.

Osim HTML forme, potrebno je izraditi i PHP datoteku naziva **skripta.php** koja će služiti za obradu podataka unesenih u HTML formi i njihov prikaz na stranici dojmova.

KOMUNIKACIJA S KORISNIKOM – KREIRANJE HTML FORME

Svaka HTML forma stvara se HTML oznakom `<form>`, te obavezno sadrži attribute *name*, *method* i *action*. Atribut *name* definira naziv forme. Obradu forme na serverskoj strani definiraju atributi *method* i *action*. Atribut *method* određuje tip HTTP zahtjeva (POST ili GET metoda), dok se atributom *action* specificira kako će biti obrađeni podaci iz forme. U sklopu laboratorijske vježbe podaci iz forme bit će obrađeni pomoću PHP skripte naziva *skripta.php*.

Prilikom kreiranja forme potrebno je pridržavati se sljedećih zahtjeva:

- web stranica **dojmovnik.html** koja sadrži formu mora biti stilski uređena pomoću vanjske CSS datoteke **style.css** kreirane u sklopu prve laboratorijske vježbe kojom su stilizirane web stranice **index.html** i **onama.html**, te mora imati isti raspored područja (zaglavlje, tijelo stranice, prostor za navigaciju i podnožje) kao i sve ostale stranice web sjedišta
- formu je potrebno smjestiti u tijelo stranice
- vrijednost atributa *action* mora biti ime PHP skripte (*skripta.php*) koja služi za obradu podataka poslanih iz forme (paziti na pravilnu upotrebu relativnog i apsolutnog adresiranja)
- vrijednost atributa *method* mora biti POST
- svi elementi forme moraju imati definiran atribut *name* koji služi kao jedinstvena identifikacijska oznaka
- elementi koje forma mora sadržavati su:
 - kućica za unos korisnikovog imena
 - kućica za unos korisnikovog prezimena
 - padajući izbornik za odabir grada
 - kućica za unos države
 - kućica za unos e-mail adrese
 - radio gumb s opcijama kojima korisnik odabire da li želi primiti obavijesti
 - kućicu za unos višelinijanskog teksta za unos pitanja ili komentara
 - gumb za slanje upisanih podataka

Sadržaj forme potrebno je kreirati unutar oznake `<form>` koju je potrebno propisno otvoriti i zatvoriti. Unutar elementa `<form>` obavezno je definirati attribute *name*, *action* i *method*.

Primjer kreiranja HTML forme

```
<form name="dojmovnik" action="putanja_do_php_skripte" method="POST">
```

SADRŽAJ FORME

```
</form>
```

Metoda kojom se podaci šalju iz forme prema serveru definirana je atributom *method*. Za slanje podataka koristite metodu POST. Postoje dvije metode kojima web preglednik šalje podatke prema serveru – GET i POST. POST metodom se svi podaci šalju kao dio zahtjeva i nevidljivi su korisniku, dok su podaci poslani GET metodom vidljivi korisniku tako što se dodaju na kraj URL-a u adresnoj traci web preglednika. Postavlja se pitanje kada koristiti koju od metoda? Npr. ako prema serveru šaljemo podatke kao što su korisničko ime i lozinka nije nam u interesu da ti podaci budu vidljivi u web pregledniku, te je u ovom slučaju POST metoda bolji izbor. Ako šaljemo veće količine podataka POST metoda je također pravi izbor. GET metodu ćemo u pravilu koristiti kada podaci koji se šalju nisu povjerljivi i kada ih nema puno.

Za kreiranje elemenata forme najčešće se koriste oznake `<input>` i `<label>`. Oznaka `<input>` služi za kreiranje interaktivnog elementa forme sa specifičnom namjenom koja se definira atributom *type*. Na ovaj način mogu se kreirati kućice za unos jedne linije teksta, radio gumbi, označna kućica (eng.

checkbox) i gumb za slanje forme. Oznaka `<label>` koristi se za definiranje oznake `<input>`. Atribut *for* oznake `<label>` treba odgovarati atributu *id* oznake s kojom se `<label>` želi povezati.

Primjer forme

```
<form name="dojmovnik" action="putanja_do_php_skripte" method="POST">

<label for="name">Ime:</label>

<input id="name" type="text" name="username" />

</form>
```

Od ostalih oznaka koje služe za kreiranje forme spomenut ćemo još:

- `<textarea>`
 - služi za kreiranje polja za unos teksta u više linija
- `<select>`
 - služi za kreiranje izbornika
 - koristi se u kombinaciji s elementom `<option>`
- `<option>`
 - služi za grupiranje elemenata izbornika
 - koristi se kao podelement oznake `<select>`

Primjer dijela forme koji služi za kreiranje izbornika

```
<form action="trigger.php" name="myForm" id="myForm" method="POST">

  <label for="username">Korisničko ime:</label>

  <input type="text" id="username" name="username" value="">

  <label for="password">Lozinka:</label>

  <input type="password" id="password" name="password" value="">

  <label for="country">Država:</label>

  <select name="country" id="country">

    <option value="">molimo odaberite</option>

    <option value="HR">Hrvatska</option>

    <option value="DE">Njemačka</option>

    <option value="US">Sjedinjene Američke Države</option>

  </select>
```

```
<label for="note">Napomena:</label>

<textarea id="note" name="note"></textarea>

<input type="submit" value="Pošalji" class="icon">

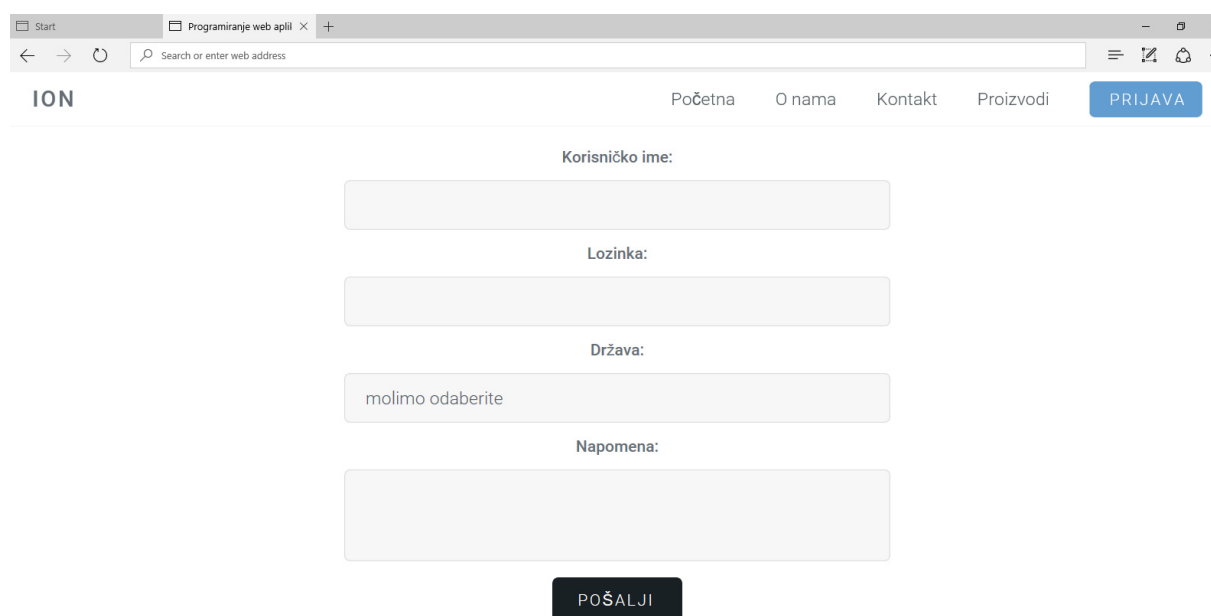
</form>
```

Na kraju forme potrebno je definirati oznaku `<input>` s vrijednošću atributa `type="submit"` kojom započinje obrada podataka iz forme. Ovime se kreira gumb kojim se podaci šalju na serversku stranu gdje će biti obrađeni PHP skriptom naziva *skripta.php*.

Primjer za izradu gumba za slanje podataka iz forme na serversku stranu.

```
<input type="submit" value="Pošalji">
```

Nakon što ste ispravno kreirali sve elemente forme u web pregledniku bi trebali dobiti prikaz sličan donjem primjeru.



The screenshot shows a web browser window with the title 'Programiranje web aplik'. The address bar contains 'Search or enter web address'. The page has a header with the logo 'ION' and navigation links: 'Početna', 'O nama', 'Kontakt', 'Proizvodi', and a blue 'PRIJAVA' button. The main content area contains a login form with the following fields and labels: 'Korisničko ime:' (username), 'Lozinka:' (password), 'Država:' (country) with a dropdown menu showing 'molimo odaberite', and 'Napomena:' (note) with a large text area. At the bottom of the form is a dark blue button labeled 'POŠALJI'.

DINAMIKA STRANICE – PHP U PRAKSI

Nakon uspješnog kreiranja HTML forme potrebno je napisati PHP skriptu kojom će se obrađivati podaci iz forme. U sklopu vježbe potrebno je kreirati skriptu koja će upisane podatke iz forme prikazati u web pregledniku.

Prilikom kreiranja PHP skripte potrebno je pridržavati se sljedećih pravila i zahtjeva:

- datoteke koje u sebi sadrže PHP kod imaju ekstenziju .php
- PHP kod uvijek započinje s <?php i završava s ?>
 - ako postoji podrška za short tag moguće je započeti pisanje koda s <?
- svaka naredba završava znakom točka-zarez (;)
- prilikom rada s varijablama važno je pridržavati se sljedećih pravila:
 - PHP varijable počinju znakom dolar (\$)
 - ime varijable mora sadržavati barem jedan znak, ne računajući znak \$
 - u imenu varijable zabranjeno je koristiti razmak i posebne znakove osim znaka podcrtavanja (_) i dolara (\$)
 - imena varijabli pisati malim slovima
 - sadržaj generiran PHP skriptom organizacijom i stilom mora odgovarati svim ostalim stranicama koje ste napravili u sklopu 1. laboratorijske vježbe; koristiti vanjsku CSS datoteku *style.css* koja je kreirana u sklopu 1. laboratorijske vježbe
 - generirani sadržaj prikazati u tijelu stranice

Prvi korak u izvršenju zadatka je razumijevanje uloge klijentske i serverske strane u prikazu sadržaja web stranica.

Na zahtjev korisnika prvo se sa servera dohvaća dokument koji sadrži formu (*dojmovnik.html*) i prikazuje na klijentskoj strani u web pregledniku. Nakon toga korisnik ispunjava HTML formu i odabirom gumba za slanje šalje podatke iz forme na server gdje se nalazi PHP skripta (*skripta.php*) koja služi za obradu podataka. Putanja do PHP skripte mora se definirati atributom *action* u elementu forme <form>. PHP skripta nakon toga dinamički kreira HTML kod koji prikazuje sadržaj forme nakon unosa podataka. U konačnici server generirani sadržaj šalje web pregledniku, a samim time na klijentskoj strani prikazuje korisniku.

HTML se izvršava na klijentskoj strani, odnosno u web pregledniku, a PHP skripta na serverskoj strani, odnosno web serveru. Kako bi provjerili da web server na kojem radite ima instaliranu podršku za PHP dovoljno je napraviti PHP datoteku sljedećeg sadržaja:

Ispis PHP konfiguracije

```
<?php
    phpinfo();
?>
```

Ukoliko podrška postoji otvaranjem kreirane PHP datoteke u web pregledniku ispisat će se razne informacije o PHP-u slično donjem primjeru. **Savjet:** ako ste kreirali spomenutu datoteku, nakon korištenja ju obrišite iz sigurnosnih razloga.

<div>phpinfo()</div> <div>localhost/dashboard/phpinfo.php</div> <div>PHP Version 5.6.21</div> <div>php</div>	
System	Windows NT DELL-PC 6.2 build 9200 (Windows 8 Professional Edition) i586
Build Date	Apr 27 2016 20:05:18
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,TS,VC11
PHP Extension Build	API20131226,TS,VC11
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring

Idući korak je razumijevanje statike i dinamike koda, te što će zapravo PHP skripta učiniti s podacima poslanima iz forme. Nakon što korisnik u formu unese podatke oni se na zahtjev šalju PHP skripti koja u našem zadatku treba kreirati sadržaj forme za prikaz korisniku u web pregledniku.

Obratimo pažnju na generiranje povratne informacije prema korisniku. Na temelju unesenih podataka želimo dobiti web stranicu koja će organizacijom i stilom odgovarati svim ostalim stranicama weba sjedišta (statični dio) i prikazivati podatke iz forme u tijelu stranice (dinamički dio). Većina PHP skripti ne sadrži isključivo PHP kod, te će se tako i u ovom slučaju datoteka *skripta.php* sastojati od statičnog HTML koda koji se ne mijenja i dinamičkog koda koji će biti generiran PHP-om.

Sjetimo se da je metoda slanja podataka iz forme prema serveru definirana u HTML formi atributom *method* u elementu `<form>`.

Primjer slanja podataka iz forme metodom POST

```
<form action="..putanja do php skripte" method="POST">

SADRŽAJ FORME

</form>
```

U navedenom primjeru svi poslani podaci iz forme spremaju se u superglobalnu varijablu `$_POST` iz koje se po potrebi mogu dohvaćati. `$_POST` je superglobalna varijabla koja je ugrađena u PHP i

direktno povezana s metodom kojom se podaci šalju iz forme. Varijabla `$_POST` je polje koje služi za pohranu više različitih podataka pod istim imenom. Podaci iz forme se u varijablu `$_POST` spremaju i dohvaćaju pomoću vrijednosti upisane pod atributom *name* svakog od elemenata forme.

Dohvaćanje podataka iz forme ostvaruje se atributom name

```
<form>
...
<input type="text" name="ime">
...
</form>
```

Podaci dohvaćeni iz HTML forme u PHP skripti se spremaju u varijable.

Primjer dohvaćanja podatka POST metodom iz HTML forme s atributom name="ime" i pohrana u PHP varijablu \$ime

```
<?php
...
$ime = $_POST['ime'];
...
?>
```

Nakon dohvata podataka poslanih iz forme potrebno je kreirati dinamički HTML kod koji će ispisivati sadržaj forme u web pregledniku. To je moguće napraviti korištenjem PHP naredbe *echo*.

Primjer korištenja naredbe echo

```
echo 'Ime koje ste unijeli je ' . $ime . '<br>';
```

U gornjem primjeru spajanje znakovnih nizova "Ime koje ste unijeli je " i imena unesenog u formi obavlja se operatorom konkatencije (znak točka (.)). Također, obratite pažnju na upotrebu znaka razmak () koji služi za odvajanje nizova.

Ako ste ispravno kreirali PHP skriptu kao rezultat bi trebali dobiti ispis u web pregledniku sličan donjem predlošku.

Vaš dojam je...

Lijep pozdrav!

Zovem se Ružica i stalno kupujem preko vašeg web shopa.

Jako sam zadovoljna vašim proizvodima i uslugom.

Hvala vam!

Ružica, Vrbovsko

UVJETI ZA PREDAJU VJEŽBE

Prilikom predaje vježbe sve kreirane stranice moraju biti funkcionalne i validne. Stranice koje moraju biti kreirane u sklopu vježbe su stranica s formom naziva *dojmovnik.html* i PHP skripta naziva *skripta.php*. Navedene stranice moraju biti oblikovane vanjskom CSS datotekom iz prve laboratorijske vježbe naziva *style.css*.