

# HTML forma, PHP

II. FAZA PROJEKTA

PROGRAMIRANJE WEB APLIKACIJA

Cilj druge faze projekta je upoznavanje s kreiranjem HTML formi, upoznavanje s osnovama jezika PHP i kreiranjem dinamičkih web stranica. U sklopu pripreme za drugu fazu projekta potrebno je proučiti osnovne HTML elemente koji služe za izradu formi, te osnovnu sintaksu jezika PHP.

U sklopu druge faze projekta potrebno je nadograditi web sjedište iz prve faze tako da se u novom HTML dokumentu naziva ***unos.html*** kreira HTML formular koji će služiti za unos novih vijesti ili proizvoda. Stranica ***unos.html*** će imati istu strukturu i navigaciju kao i dosadašnji dijelovi aplikacije. Dodati link na tu stranicu u navigaciju. 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. Dokument ***skripta.php*** treba prema predlošku dobivenog u zadatku oblikovati unos iz forme u detaljni prikaz jednog članka ili proizvoda. Dakle, **stranice koje moraju biti kreirane u sklopu druge faze su** stranica s HTML formom naziva ***unos.html*** i PHP skripta naziva ***skripta.php***. Navedene stranice moraju biti oblikovane vanjskom CSS datotekom iz prve faze projekta naziva ***style.css***. Sadržaj koji je potrebno kreirati u sklopu navedenih stranica detaljno je opisan u nastavku uputa.

## **Komunikacija S Korisnikom – Kreiranje HTML Forme**

HTML obrazac dio je web-stranice koja obuhvaća područja na kojima korisnici mogu unijeti informacije koje će vam se poslati natrag, na drugu adresu e-pošte koju navedete, u bazu podataka koju upravljate ili u drugi sustav, kao što je treća sustav za upravljanje strankama za vodeće generacijske obrasce vaše tvrtke, kao što je Salesforce.com

Obrasci uključuju gumb koji korisnik može kliknuti za slanje obrazaca; taj gumb može biti slika koju sami kreirate ili standardni gumb HTML obrasca koji se kreira kada se kreira element `<input>` i daje vrijednost tipa submit. Kada netko klikne gumb za podnošenje obrasca, sve informacije upisane u obrazac šalju se na URL koji navedete u atributu akcije elementa `<form>`. Taj bi URL trebao upućivati na određenu skriptu koja će obraditi vaš obrazac, slanje sadržaja obrasca putem e-pošte ili izvođenje drugog koraka u interaktivnom procesu (kao što je traženje rezultata iz tražilice ili stavljanje stavki u košaricu za kupnju).

## Stvaranje Obrasca (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. Podaci iz forme bit će obrađeni pomoću PHP skripte naziva skripta.php.

Prilikom kreiranja forme potrebno je pridržavati se sljedećih zahtjeva:

- web stranica unos.html koja sadrži formu mora biti stilski uređena pomoću vanjske CSS datoteke style.css kreirane u sklopu prve faze projekta, 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 naslova vijesti
  - kućicu za unos višelinijskog teksta za unos kratkog sažetka vijesti
  - kućicu za unos višelinijskog teksta za unos teksta vijesti o padajućim izborniku u kojem je moguće odabrati kategoriju vijesti (kategorije odredite samostalno)
  - padajuća lista za odabir kategorije vijesti
  - element za odabir slike
  - okvir za izbor (engl. checkbox) da li da se obavijest prikazuje na stranici
  - 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.

```
<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.

Atribut `action` navodi adresu za slanje podataka obrasca. Ovdje imate dvije mogućnosti:

- Na web poslužitelj možete upisati mjesto skripte za obradu obrazaca, a podaci obrasca će se zatim poslati u tu skriptu. To je daleko najčešći scenarij.

```
<form name = "my_form" method = "POST" action = "skripta.php">
```

- Možete upisati `mailto:` slijedi vaša adresa e-pošte, a podaci obrasca bit će vam poslani izravno kad god netko popuni obrazac. Međutim, ovaj pristup je u potpunosti ovisan o konfiguraciji korisnikovog računala.

```
<form name = "my_form" method = "POST" action = "mailto: me@mysite.com">
```

Neki od novih atributa koji se koriste prilikom kreiranja formi u standardu HTML 5, a koje možete iskoristiti prilikom izrade projekta, su:

- `autocomplete`
  - atribut koji služi za postavljanje mogućnosti automatske nadopune teksta
  - koristi se s oznakama `<form>` i `<input>`
  - moguće vrijednosti atributa su `on` i `off`
  - u nekim web preglednicima potrebno je uključiti opciju samodopune kako bi navedeni atribut djelovao

- autofocus
  - atribut koji stavlja kursor u oznaku `<input>` u kojoj se koristi, tj. prilikom otvaranja stranice kursor se postavlja u taj dio forme

## Imenovanje Elemenata Obrasca (atribut name)

Bez obzira na vrstu elementa obrasca, morate dati naziv podacima koje prikuplja. Možete koristiti bilo koje ime koje želite za svaku stavku unosa, sve dok je svaka na obrascu drugačija (osim u slučaju radio buttona). Kada obrazac obrađuje pozadinska skripta, svaka stavka podataka identificira se po imenu. Ovo ime postaje varijabla, koja je ispunjena vrijednošću. Vrijednost je ili ono što je korisnik unio u obrazac ili vrijednost povezana s elementom koji je korisnik odabrao.

## Input Kontrole Obrasca

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.

## Check Boxes

Osim tekstnog polja, jedan od najjednostavnijih tipova ulaza je potvrdni okvir koji se pojavljuje kao mali kvadrat. Korisnici mogu kliknuti potvrdne okvire da bi odabrali ili poništili odabir jedne ili više stavki u grupi.

Check Box ima sljedeća četiri svojstva:

- *name* je naziv potvrdnog okvira kao i ime objekta.
- *value* je "true" vrijednost za potvrdni okvir - obično uključena. Tu vrijednost koriste programi na strani poslužitelja kako bi označili je li potvrđen okvir za izbor. U JavaScriptu biste umjesto toga trebali koristiti označeno svojstvo.
- *defaultChecked* je zadani status potvrdnog okvira, dodijeljen atributom *checked* u HTML-u.

- Checked je trenutna vrijednost. Ovo je logička vrijednost: true za označeno i false za neoznačeno.

```
<input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
<input type="checkbox" name="vehicle2" value="Car"> I have a car<br>
<input type="checkbox" name="vehicle3" value="Boat" checked> I have a boat<br>
```

## Radio Buttons

Radio buttons, za koje se može odabrati samo jedan izbor u isto vrijeme, gotovo su jednako jednostavni za implementaciju kao i potvrdni okviri. Najjednostavnije korištenje radio gumba je za pitanja da / ne ili za pitanja kada se može odabrati samo jedan odabir.

Da biste stvorili radio button, koristite type = "radio" i svakoj opciji dodijelite vlastitu oznaku <input>. Upotrijebite isti naziv za sve radio gumbe u grupi:

Radio buttoni su slični potvrdnim okvirima, osim što cijela grupa dijeli jedno ime i jedan objekt. Mogu sadržavati sljedeća svojstva:

- name je zajedničko ime za radio gumbe.
- length je broj radio gumba u grupi.
- value je vrijednost dodijeljena gumbu.
- defaultChecked označava vrijednost atributa za provjeru i zadano stanje gumba.
- Checked je trenutno stanje

```
<input type="radio" name="gender" value="male"> Male<br>
<input type="radio" name="gender" value="female"> Female
```

## Selection Lists

Padajuća lista stvara se sa <select> </select> tagom, između kojih se nalaze elementi liste napisani unutar <option></option> taga.

<select> može sadržavati sljedeća svojstva:

- name je naziv liste za odabir.
- size je broj opcija na popisu koji su vidljivi.
- selectedIndex vraća vrijednost indeksa trenutno odabrane stavke.

```
<select name=auto size="2">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

## Text Fields, Text Areas, i ostali input tipovi

### Text

Koristi se kada se želi korisniku dopustiti da unese samo jedan redak teksta.

```
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
```

### Textarea

Kada želite omogućiti više redaka teksta u jednoj stavci unosa, upotrijebite oznake <textarea></textarea> da biste stvorili tekstualno područje umjesto samo tekstualnog polja. Svaki tekst koji uključite između ove dvije oznake prikazuje se kao zadani unos u tom okviru.

Atributi rows i cols kontroliraju broj redaka i stupaca teksta u okviru za unos.

```
<textarea rows="4" cols="50" name="poruka"> </textarea>
```

### Ostali tipovi

- type = "email"  

```
<input type="email" name="email">
```
- type = "file"  

```
<input type="file" name="myFile">
```

- type = "number"  
`<input type="number" name="quantity" min="1" max="5">`
- type = "range"  
`<input type="range" name="points" min="0" max="10">`
- type = "url"  
`<input type="url" name="homepage">`
- type = "color"  
`<input type="color" name="favcolor" value="#ff0000">`
- type = "hidden"  
`<input type="hidden" id="custId" name="custId" value="3487">`
- type = "password"  
`<input type="password" name="psw">`

## Slanje Podataka

Prilikom kreiranja forme potrebno je pridržavati se sljedećih zahtjeva:

- web stranica unos.html koja sadrži formu mora biti stilski uređena pomoću vanjske CSS datoteke style.css kreirane u sklopu prve faze projekta, 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 naslova vijesti
  - kućicu za unos višelinijskog teksta za unos kratkog sažetka vijesti
  - kućicu za unos višelinijskog teksta za unos teksta vijesti o padajući izbornik u kojem je moguće odabrati kategoriju vijesti (kategorije odredite samostalno)
  - padajuća lista za odabir kategorije vijesti
  - element za odabir slike
  - okvir za izbor (engl. checkbox) da li da se obavijest prikazuje na stranici
  - gumb za slanje upisanih podataka

```

<form action="skripta.php" method="POST">
    <div class="form-item">
        <label for="title">Naslov vijesti</label>
        <div class="form-field">
            <input type="text" name="title" class="form-field-textual">
        </div>
    </div>
    <div class="form-item">
        <label for="about">Kratki sadržaj vijesti (do 50
znakova)</label>
        <div class="form-field">
            <textarea name="about" id="" cols="30" rows="10" class="form-
field-textual"></textarea>
        </div>
    </div>
    <div class="form-item">
        <label for="content">Sadržaj vijesti</label>
        <div class="form-field">
            <textarea name="content" id="" cols="30" rows="10"
class="form-field-textual"></textarea>
        </div>
    </div>
    <div class="form-item">
        <label for="pphoto">Slika: </label>
        <div class="form-field">
            <input type="file" accept="image/jpg,image/gif"
class="input-text" name="pphoto"/>
        </div>
    </div>
    <div class="form-item">
        <label for="category">Kategorija vijesti</label>

```

```

        <div class="form-field">
        <select name="category" id="" class="form-field-textual">
        <option value="sport">Sport</option>
        <option value="kultura">Kultura</option>
        </select>
        </div>
    </div>
    <div class="form-item">
        <label>Spremiti u arhivu:
        <div class="form-field">
            <input type="checkbox" name="archive">
        </div>
        </label>
    </div>
    <div class="form-item">
        <button type="reset" value="Poništi">Poništi</button>
        <button type="submit" value="Prihvati">Prihvati</button>
    </div>

</form>

```

Naslov vjesti

Kratki sadržaj vjesti (do 50 znakova)

Sadržaj vjesti

Slika:

No file chosen

Kategorija vjesti

Sport

Spremiti u arhivu:

☐

## PHP

Nakon uspješnog kreiranja HTML forme potrebno je napisati PHP skriptu kojom će se obrađivati podaci iz forme. U sklopu druge faze 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; koristiti vanjsku CSS datoteku style.css koja je kreirana u sklopu prve faze projekta
  - 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 (unos.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. Za potrebe projekta koristit će se XAMPP web server.

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>`.

```
<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.

Pretpostavimo da obrazac sadrži vrijednost tekstualnog polja pod nazivom title koji izgleda ovako:

```
<input type="text" name="title" class="form-field-textual">
```

Nakon slanja ovog obrasca možete referencirati vrijednost tog tekstualnog polja na sljedeći način:

```
$_POST['title']
```

Naravno, zbog praktičnosti, ništa vas ne sprječava da najprije dodijelite ovu vrijednost drugoj varijabli, na primjer:

```
$title=$_POST['title'];
```

## Korištenje isset() funkcije

Function `isset()` određuje je li varijabli dodijeljena vrijednost.

```
isset ( mixed $var [, mixed $... ] ) : bool
```

Vraća TRUE ako je varijabla postavljena i sadrži vrijednost različitu od null vrijednosti i FALSE ako ne.

```
if (isset($_POST['title']))
{
    $title=$_POST['title'];
}
```

## Ispis

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`

```
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 konkatenacije (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 predlošku kojeg ste dobili u zadatku.

```
<section role="main">
  <div class="row">
    <p class="category"><?php
      echo $category;
    ?></p>
    <h1 class="title"><?php
      echo $title;
    ?></h1>
    <p>AUTOR:</p>
    <p>OBJAVLJENO:</p>
  </div>
  <section class="slika">
    <?php
      echo "<img src='img/$image'";
    ?>

  </section>
  <section class="about">
    <p>
      <?php
        echo $about;
      ?>
    </p>

  </section>
  <section class="sadrzaj">
    <p>
      <?php
        echo $content;
      ?>
    </p>
  </section>

</section>
```

## Sport

### BUCSKI PREOKRETOM PROTIV RAPTORSA OBRANILI PREDNOST DOMAĆEG TERENA

AUTOR:

OBJAVLJENO:



*Lopez i Antetokounmpo sjajnim partijama osigurali prvu pobjedu u seriji*

Milwaukee Bucks pobjedili su Toronto Raptorse 108-100 u prvoj utakmici finala Istočne konferencije u NBA ligi. Raptors su pune tri četvrtine imali utakmicu pod kontrolom i bili u vodstvu punih 37 minuta utakmice. Međutim, u posljednjoj četvrtini uslijedio je preokret Bucks, i to na krilima Brooka Lopeza koji je utakmicu završio s 29 poena i 11 skokova. Lopez je u posljednjoj četvrtini ubacio 13 poena. Dvije minute i dvadeset sekundi prije kraja Lopez je zakucavanjem doveo Buckse u vodstvo koje do kraja utakmice nisu ispustili. Posljednju četvrtinu Raptors su dobili rezultatom 32-17, a utakmicu su završili serijom 10-0. Giannis Antetokounmpo ostvario je učinak od 24 poena, 11 skokova i šest asistencije. Treći najbolji strijelac Bucks bio je Malcolm Brogdon s 15 poena. Kod Raptorsa, Kawhi Leonard ubacio je 31 poena uz devet skokova, dok je poen manje postigao Kyle Lowry. - Ubila nas je posljednja četvrtina. Nadigrali su nas. Podigli su agresivnost i ubacili neke važne šuteve. Priliku smo imali i sada iz toga moramo nešto naučiti, rekao je nakon utakmice Lowry. Druga utakmica na rasporedu je u noći s petka na subotu.

Copyright © HANZA MEDIA d.o.o. Sva prava pridržana.