

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET**

Sveučilišni studij

**WEB APLIKACIJA ZA VLASNIKE KUĆNIH
LJUBIMACA**

Završni rad

Monika Čivić

Osijek, 2014.

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom studiju

Osijek,

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Monika Čivić		
Studij, smjer:	Prediplomski studij, računarstvo		
Mat. br. studenta, godina upisa:	0165049428, 2011.		
Mentor:	Doc.dr.sc. Krešimir Nenadić		
Sumentor:			
Naslov završnog rada:	Web aplikacija za vlasnike kućnih ljubimaca		
Primarna znanstvena grana rada:			
Sekundarna znanstvena grana (ili polje) rada:			
Predložena ocjena završnog rada:			
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: Postignuti rezultati u odnosu na složenost zadatka: Jasnoća pismenog izražavanja: Razina samostalnosti:		
Potpis sumentora:		Potpis mentora:	
Dostaviti:			
1. Studentska služba			
<div style="text-align: right;">Potpis predsjednika Odbora:</div>			
Dostaviti:			
1. Studentska služba			



Sveučilište Josipa Jurja Strossmayera u Osijeku

ETFOS

ELEKTROTEHNIČKI FAKULTET OSIJEK



IZJAVA O ORIGINALNOSTI RADA

Osijek,

Ime i prezime studenta:

Monika Čivić

Studij :

Preddiplomski studij

Mat. br. studenta, godina upisa:

0165049428, 2011.

Ovom izjavom izjavljujem da je rad pod nazivom:
Web aplikacija za vlasnike kućnih ljubimaca

izrađen pod vodstvom mentora

Doc.dr.sc. Krešimira Nenadića

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD	1
1.1. Zadatak završnog rada	1
2. KLIJENTSKE TEHNOLOGIJE	2
2.1. Hypertext Markup Language	2
2.1.1. Struktura i primjena HTML-a	2
2.2. Cascading Style Sheets	4
2.2.1. Radni okvir za CSS	5
2.2.2. Struktura i primjena CSS-a	5
2.3. JavaScript.....	7
2.3.1. Primjena JavaScripta	7
2.3.2. AJAX.....	9
3. POSLUŽITELJSKE TEHNOLOGIJE	12
3.1. PHP Hypertext Preprocessor	12
3.1.1. Primjena i primjeri	12
3.2. Baza podataka - MySQL	15
3.2.1. Primjena i primjeri	15
4. WEB APLIKACIJA	18
4.1. Prezentacijski sloj	18
4.2. Aplikacijski i podatkovni sloj.....	19
4.2.1. Sustav za registraciju i prijavu	20
4.2.2. Rad s ljubimcima.....	21
4.2.3. Preporuke i podsjetnici.....	22
4.2.4. Ostali dijelovi aplikacije.....	24
5. TESTIRANJE.....	25
6. ZAKLJUČAK	28
Literatura	29
Sažetak	30
Abstract	31
Životopis.....	32

1. UVOD

Web aplikacije su računalni programi koji korisnicima pružaju svoje usluge na mreži. Takve aplikacije pokreće web preglednik čijim se posredstvom vrši izmjena podataka između korisnika i poslužitelja. Velika prednost web aplikacija, u odnosu na standardne klijent – poslužitelj aplikacije, je mogućnost održavanja i ažuriranja njenih funkcija bez distribucije i instalacije na tisuće korisničkih računala. Razlog tomu je što web aplikacija umjesto izvršne datoteke generira skup dokumenata koji se mogu koristiti u web pregledniku. Ono po čemu se web aplikacije razlikuju od statičnih web stranica je to da se pri njihovoj izradi isprepliću klijentske i poslužiteljske web tehnologije. Klijentske tehnologije poput HTML-a, CSS-a i JavaScripta služe za prezentaciju i interaktivnost stranica. S druge strane, poslužiteljske tehnologije omogućuju dinamički rad web stranice generiranjem sadržaja podacima iz baze. Poslužiteljske tehnologije obrađene u poglavljima koji slijede su programski jezik PHP te MySQL baza podataka.

Kao praktična primjena navedenih načela i tehnologija, izrađena je jednostavna web aplikacija nazvana *APPet*. Svrha aplikacije je pomoći vlasnicima kućnih ljubimaca u vođenju evidencije i pravilnoj brizi o svojim ljubimcima. Takva aplikacija je kombinacija HTML, CSS, JavaScript i PHP kôda, ali i baze podataka. Kroz primjere iz izvornog kôda aplikacije objašnjena je uloga pojedine web tehnologije u njenom pravilnom radu. Osnovna svojstva aplikacije su prijava i registracija korisnika i sustav za manipulaciju ljubimcima. Glavne sastavnice potonjeg sustava su dodavanje i brisanje ljubimaca, izmjena podataka te izrada preporuka i podsjetnika na temelju određenih karakteristika ljubimca. Nakon završetka izrade aplikacije obavljeno je i testiranje kako bi se dobio uvid u sve nedostatke, moguće previde, ali i vidjelo ima li prostora za proširenje aplikacije dodatnim funkcijama.

1.1. Zadatak završnog rada

Kao zadatak završnog rada napravljena je web aplikacija pomoću koje vlasnik kućnog ljubimca, nakon prijave u sustav, unosi podatke o jednom ili više ljubimaca. Prema unesenim podacima i unaprijed zadanim kriterijima, aplikacija predlaže plan prehrane i aktivnosti ljubimca. U aplikaciju su ugrađena i upozorenja za hranjenje ljubimca i, gdje je potrebno, cijepljenje.

2. KLIJENTSKE TEHNOLOGIJE

Klijentsko-poslužiteljski način rada karakterističan je za gotovo sva web mjesta na današnjem Internetu. Klijenti su uređaji koji imaju mogućnost pregledavanja web stranica i služe za traženje i obradu informacija, odnosno slanje zahtjeva poslužitelju. S druge strane, poslužitelji su uređaji na kojima se nalaze razne informacije, a potrebno ih je pružiti klijentu kada to zatraži. Na temelju klijentsko-poslužiteljskog odnosa nastala je podjela web tehnologija ovisno o tome obavljaju li zadatke na poslužiteljskoj (engl. *server-side*) ili klijentskoj (engl. *client-side*) strani. Klijentskim tehnologijama pripadaju tehnologije za opis sadržaja, prezentaciju i interaktivnost stranice.

2.1. Hypertext Markup Language

HyperText Markup Language, ili skraćeno HTML, opisni je, prezentacijski jezik za izradu web stranica. Hipertekstualni dokumenti koje otvaraju preglednici, stvaraju se upravo pomoću HTML jezika. Takvi dokumenti zapravo su obične tekstualne datoteke s ekstenzijama *.html* ili *.htm*. HTML čini osnovnu građevnu strukturu svake web stranice, bila ona statična ili dinamična, pa tako i osnovnu strukturu web aplikacije za vlasnike kućnih ljubimaca. Pomoću HTML-a, oblikovan je sadržaj web aplikacije i stvorene poveznice na ostale hipertekstualne dokumente.

Prilikom oblikovanja sadržaja aplikacije, korištene su mogućnosti poput umetanja slika i objekata te izrade obrazaca (engl. *forms*). Isto tako, upotrijebljene su i oznake za različite vrste naslova, podnaslova, odlomaka, popisa i slično. Važno je naglasiti da, iako pruža brojne mogućnosti, HTML nije programski jezik. Razlog tomu je što se u HTML-u ne mogu izvršiti čak ni osnovne računske operacije kao što su zbrajanje i oduzimanje. Prema tome, HTML nije jezik pomoću kojeg su riješene programske zadaće aplikacije, već samo jezik koji koristi sintaksu sličnu engleskom jeziku za opisivanje sadržaja stranica na webu.

2.1.1. Struktura i primjena HTML-a

Osnovne jedinice HTML dokumenta su HTML elementi. Elementi imaju određene uloge u dokumentu, a sastoje se od para HTML oznaka i dijela kôda koji se nalazi između njih. Taj dio kôda naziva se sadržajem elementa. HTML dokumenti počinju oznakom tipa dokumenta „`<!DOCTYPE HTML>`“ i sastoje se od zaglavlja (unutar oznaka „`<head>`“) i tijela (unutar „`<body>`“ oznaka). U izradi aplikacije *APPet* korišteni su mnogobrojni HTML elementi s

različitim funkcijama. Ipak, najčešće su korištene oznake: „<div>“ za definiranje odjeljaka, „<a>“ za stvaranje poveznica, „<form>“ za izradu obrazaca, te „“ za kreiranje popisa. Često korišteni atributi, koji sadrže dodatne informacije o pojedinom elementu, su „class“ i „id“. Oni služe i za lakšu stilizaciju u CSS-u. U primjeru (Sl.2.1.) je prikazan način primjene HTML jezika za izradu početne stranice aplikacije, ali bez zaglavlja i podnožja.

```
1 <div class="headline">
2   <div class="row">
3     <div class="text-center">
4       <h1><?php echo "Dobrodošli u APPet zajednicu!"; ?></h1>
5     </div>
6   </div>
7   <div class="row">
8     <div class="text-center">
9       
10    </div>
11  </div>
12  <div class="row">
13    <div class="text-center">
14      <ul class="button-group radius">
15        <li>
16          <a href="<?php echo BASE_URL;?>signup/" class="button text-
17            center">Registracija</a>
18        </li>
19        <li>
20          <a href="<?php echo BASE_URL;?>login/" class="button no2
21            text-center">Prijava me</a>
22        </li>
23      </ul>
24    </div>
25  </div>
26 </div>
27 </div>
```

Sl. 2.1. *Isječak kôda iz dokumenta koji predstavlja početnu stranicu.*

Primjer sa slike 2.1. pokazuje da HTML oznake čine osnovnu građevnu strukturu početne, ali i svih ostalih stranica aplikacije. Unatoč tomu, bez umetanja CSS stilova, stranice su gotovo neupotrebljive. To se može vidjeti i na slici 2.2., koja prikazuje izgled dokumenta sa slike 2.1. u web pregledniku kada na njega nisu primijenjena CSS svojstva. Dakle, dizajn web stranice nemoguć je samo uz korištenje HTML-a jer čak ni jednostavnim elementom kao što je slika nije moguće adekvatno manipulirati.

Dobrodošli u APPet zajednicu!



- [Registracija](#)
- [Prijavi me](#)

Sl. 2.2. Prikaz primjera sa slike 2.1. u web pregledniku.

2.2. Cascading Style Sheets

Cascading Style Sheets, poznatiji pod kraticom CSS, stilski je jezik i služi za definiranje različitih stilova za prikaz web stranica. Pomoću njega određuje se način na koji će dokumenti napisani u HTML-u biti prezentirani korisniku. Uz pomoć CSS-a moguće je odrediti veličinu, boju, vrstu fonta, ali i mnoga druga obilježja pojedinih HTML elemenata. CSS datoteke su obične tekstualne datoteke s ekstenzijom *.css*. Ipak, CSS kôd nije potrebno uvijek pisati u zasebnoj datoteci, već je to moguće i unutar HTML datoteke što je rjeđe primjenjivan način. Razlog tomu je što je CSS nastao kako bi se sadržaj dokumenta tj. web stranice, odvojio od njenog prezentacijskog dijela. Takav način rada poboljšava dostupnost samog sadržaja te daje veću fleksibilnost i kontrolu nad karakteristikama izgleda stranice. Isto tako, CSS omogućuje dodavanje jednakih obilježja većem broju različitih stranica pa nema potrebe za kopiranjem pojedinih dijelova kôda što uvelike skraćuje proces izrade i prepravljanja. Za svaki HTML element, CSS sadrži popis svojstava za oblikovanje. Sintaksa CSS-a sastoji se od brojnih

ključnih riječi na engleskom jeziku te ih se lako može povezati sa stilskim značajkama za koje su namijenjeni.

2.2.1. Radni okvir za CSS

CSS radni okviri su unaprijed pripremljeni programski radni okviri koji pružaju mogućnost lakšeg, više standardiziranog web dizajniranja koristeći CSS. Većina takvih okvira sadrži bar mrežu (engl. *grid*) za lakšu organizaciju objekata na stranici. Kako bi izgled aplikacije za vlasnike kućnih ljubimaca bio što bolji i pružio dobro korisničko iskustvo (engl. *user experience*), pri dizajniranju aplikacije korišten je besplatni radni okvir *Foundation*. To je zapravo skup alata za izradu web stranica i aplikacija te sadrži predloške zasnovane na HTML-u, CSS-u i JavaScript skriptama. Aplikacije izrađene pomoću *Foundationa*, pa tako i *APPet*, prikladne su za prikaz na tabletima i mobilnim uređajima jer je mreža koju *Foundation* sadrži u potpunosti responzivna. Prednost radnih okvira je što postoji i mogućnost izmjene gotovih klasa. Tako je za potrebe aplikacije i bolju prilagodbu zamislima dizajnera korištena dodatna CSS datoteka za promjenu unaprijed zadanih *Foundation* stilova.

2.2.2. Struktura i primjena CSS-a

CSS datoteke sa stilovima primijenjenim na web aplikaciju (SI.2.3.) sastoje se od jednog ili više selektora i deklaracijskih blokova s različitim svojstvima. Selektorima je određeno na koji je dio HTML dokumenta potrebno primijeniti pojedine stilske značajke. Druga važna instanca CSS kôda su deklaracijski blokovi. Deklaracijski blokovi sastoje se od deklaracija svojstava unutar vitičastih zagrada. Svaka se deklaracija sastoji od svojstva, dvotočke i vrijednosti. Sva svojstva određena su standardom za CSS i svako od njih ima nekoliko mogućih vrijednosti. Nasljeđivanje je također jedna od važnih značajki CSS-a, a zasniva se na principu veze između roditelja i djece. Nasljeđivanje je koncept na temelju kojega se određena svojstva primjenjuju, ne samo na elemente za koja su zadana, već i na sve elemente unutar njih.

```
1  h1 {  
2      font-family: 'Poiret One', sans-serif;  
3  }  
4  .headline {  
5      margin-top: 30px;  
6  }  
7  .button {  
8      background-color: black;  
9      padding: 15px 30px;  
10     display: inline-block;  
11     font-family: 'Poiret One', sans-serif;
```

```

12     font-weight: bold;
13     font-size: 16px;
14     color: white;
15     border: 0px;
16 }
17 .no2 {
18     background-color: #FF3B19;
19 }
20 #main-img {
21     margin: 10px 0 50px 0;
22 }

```

Sl. 2.3. Dio kôda iz datoteke sa stilovima koji se odnose na početnu stranicu iz primjera sa slika 2.1. i 2.2.

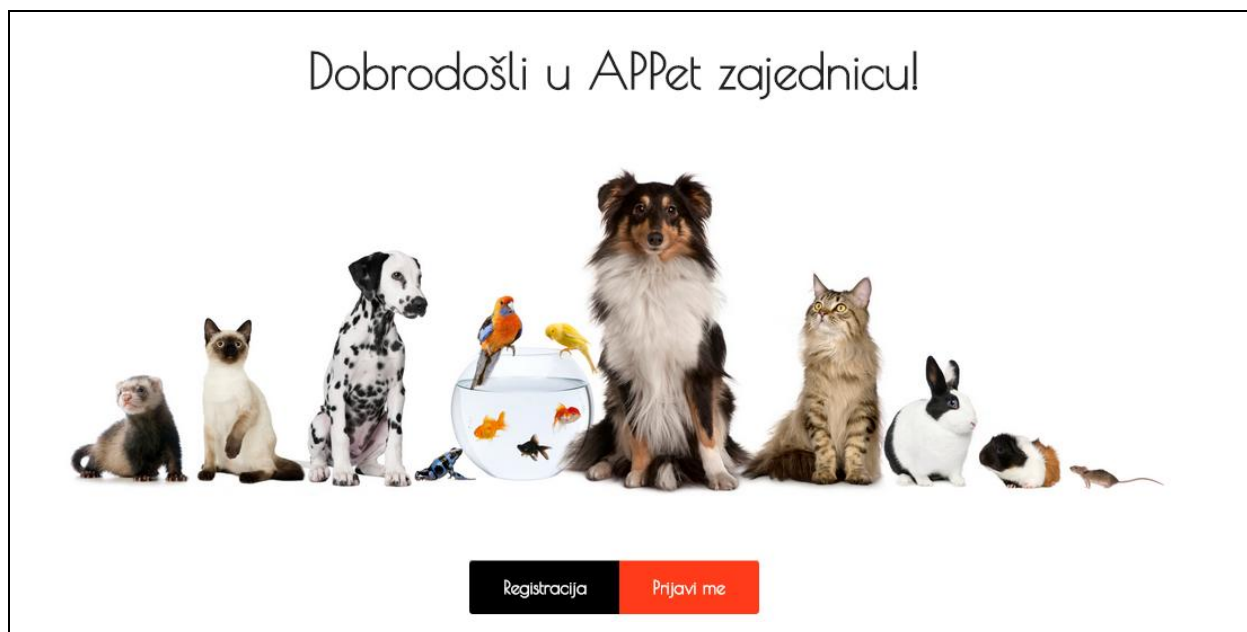
Iako postoje tri različita načina na koja se CSS stilovi mogu primijeniti na HTML, u *APPet* aplikaciju umetnuti su kao vanjske datoteke pomoću „<link>“ oznaka u zaglavlju HTML dokumenta. Najprije su dodani gotovi, *Foundation* stilovi, a zatim i datoteka s vlastitim stilovima. Kad je riječ o dizajnu aplikacije, uz slaganje boja i funkcionalnu organizaciju stranice, važna je i tipografija. Iz tog razloga, dodane su i poveznice na besplatne *GoogleFonts* fontove. Način na koji su umetnuti CSS stilovi prikazan je na slici ispod (Sl.2.4.).

```

1 <head>
2   <link href="<?php echo BASE_URL;?>css/normalize.css" rel="stylesheet"
3     media="screen">
4   <link href="<?php echo BASE_URL;?>css/foundation.css" rel="stylesheet"
5     media="screen">
6   <link href="<?php echo BASE_URL;?>css/foundation-icons.css"
7     rel="stylesheet" media="screen">
8   <link href="<?php echo BASE_URL;?>css/my-styles.css" rel="stylesheet"
9     media="screen">
10  <link href='http://fonts.googleapis.com/css?family=Open+Sans&subset=
11    latin,latin-12ext' rel='stylesheet' type='text/css'>
12  <link href='http://fonts.googleapis.com/css?family=Poiret+One&subset=
13    latin,latin15-ext' rel='stylesheet' type='text/css'>
14 </head>

```

Sl. 2.4. Način umetanja CSS stilova i fontova u aplikaciju.



Sl. 2.5. Izgled primjera sa slike 2.1. nakon umetnutog CSS-a.

2.3. JavaScript

JavaScript, ili skraćeno JS, dinamički je računalni programski jezik. Najviše se koristi u sklopu web preglednika. Implementacija JS-a pruža mogućnost pisanja skripti na klijentskoj strani za stvaranje bolje komunikacije s korisnikom. Uz JS moguće je upravljati preglednikom i mijenjati sadržaje dokumenata prikazanih u pregledniku. Nadalje, JavaScript se koristi i na strani poslužitelja za mrežno programiranje, razvoj igara i izradu računalnih i mobilnih aplikacija. JavaScript trenutno je najpopularniji skriptni jezik na Internetu, a podržavaju ga svi poznatiji preglednici. Javno je raspoloživ jezik pa za njegovo korištenje nije potrebna licenca.

2.3.1. Primjena JavaScripta

JavaScript funkcije moguće je pisati u HTML dokumentu između oznaka „<script></script>“ ili u vanjskoj datoteci s ekstenzijom `.js`. Iako JavaScript sadrži brojne karakteristike koje mogu povećati interaktivnost web stranica, aplikacija *APPet* ne sadrži mnogo JavaScript kôda. Međutim, koristi neke JavaScript funkcije koje se pozivaju na događaje, odnosno na promjene na web stranici. HTML događaji na koje se izvršavaju JS skripte mogu biti bilo što što čine preglednik ili korisnik. Neki od događaja korištenih u aplikaciji su „onclick“ i „onload“, no postoji i nekoliko drugih događaja. Dakle, odabrani su elementi na koje treba primijeniti JS te im dodani događaji i pozivi funkcija. Primjer takvog korištenja JS-a jest na stranici s popisom ljubimaca. Naime, popis ljubimaca, osim imena ljubimaca, sadrži i gumb za

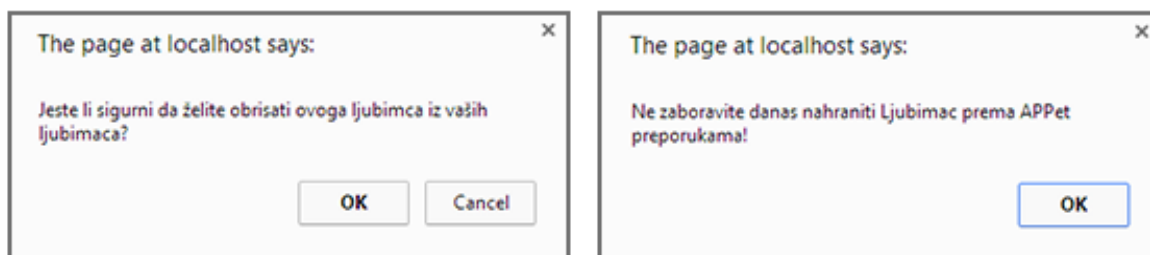
njihovo uklanjanje s popisa. Kako korisnik ne bi nehotice uklonio ljubimca, klikom na gumb za brisanje poziva se događaj koji traži potvrdu te radnje. U suprotnom, radnja se neće izvršiti (Sl.2.6.).

```
1 <li>
2   <a href="<?php echo BASE_URL . 'pets/' . $pet['pet_id'] . '/'; ?>"
3   class="button no7 radius"><?php echo $pet['pet_name']; ?> </a>
4   <a href="<?php echo BASE_URL . 'pets/delete_pet.php?id=' .
5   $pet['pet_id']; ?>" class="button no8 radius"
6   onclick="javascript:return
7   window.confirm("Jeste li sigurni da želite ukloniti ovog ljubimca iz
8   vaših ljubimaca?");"><i class="fi-x"></i></a>
9 </li>
```

Sl. 2.6. Dio kôda za ispis jednog ljubimca na popis.

Najčešće JS funkcije s kojima se korisnici susreću pri korištenju web preglednika su različiti skočni prozori i kolačići. Skočni prozori (engl. *pop-ups*) pripadaju JS objektu *window* kojeg podržavaju svi moderni preglednici, a predstavlja prozor preglednika. Postoje tri osnovne vrste skočnih prozora u JavaScriptu, a to su prozor upozorenja (engl. *alert box*), prozor za potvrdu (engl. *confirm box*) i odzivnik (engl. *prompt*). Prve dvije funkcije dio su aplikacije za vlasnike kućnih ljubimaca.

Prozor upozorenja (Sl.2.7., desno) koristi se kada je potrebno osigurati da neka informacija dođe do korisnika. U tom slučaju, korisnik mora kliknuti *OK* ne bi li nastavio s daljnjim radom. Poziva se pomoću ključnih riječi *window.alert()*. Taj prozor korišten je kao podsjetnik za hranjenje i cijepljenje ljubimca. Prozor za potvrdu (Sl.2.7., lijevo) koristi se kada je potrebna korisnička verifikacija ili potvrda. Kada se takav prozor pojavi, korisnik treba kliknuti ili *OK* ili *Cancel* kako bi nastavio s radom. U slučaju da odabere *OK* prozor vraća vrijednost *true*, a u slučaju da odabere *Cancel* vraća *false*. Naredba za pozivanje je *window.confirm()*. Upravo takav prozor traži potvrdu korisnika aplikacije prije brisanja određenog kućnog ljubimca s popisa ljubimaca.



Sl. 2.7. Primjeri skočnih prozora korištenih u aplikaciji.

Osim skočnih prozora JavaScript je upotrijebljen za stvaranje kolačića (engl. *cookies*). Kolačići su podaci pohranjeni u malim tekstualnim datotekama na računalu. Kada web poslužitelj pošalje web stranicu pregledniku, veza se prekida i poslužitelj „zaboravlja“ sve podatke o korisniku. Kolačići su osmišljeni kako bi riješili problem „pamćenja“ informacija o korisniku. Kada korisnik posjeti web stranicu njegovo ime može biti pohranjeno u kolačiću. Sljedeći put kada taj isti korisnik posjeti stranicu, kolačić će se „sjećati“ njegova imena. Uz pomoć JS-a moguće je kreirati, čitati, mijenjati i brisati kolačiće uz pomoć svojstva *document.cookie*.

Za korištenje kolačića potrebne su najmanje tri funkcije i to jedna za postavljanje kolačića, druga za dobavljanja kolačića te posljednja za njihovu provjeru. Jedna od tih funkcija nalazi se na slici 2.8. *APPet* aplikacija koristi kolačiće prilikom alarmiranja korisnika o hranjenju. Kada korisnik pristupi podacima o ljubimcu, pojavljuje se skočni prozor upozorenja (Sl.2.7., desno). Kako se to ne bi događalo svaki put kad korisnik pristupi ljubimcu, kolačići pamte njegov posjet. Tim kolačićima dodano je i vrijeme isteka od dvanaest sati kako se ne bi pojavljivali više od dva puta na dan. Na isti način, korisnik je, u slučaju da posjeduje psa, upozoren na cijepljenje istog, ali taj kolačić ima duže vremensko trajanje.

```
1 //funkcija za postavljanje kolačića
2 function setCookie(cname, cvalue, exdays) {
3     var d = new Date();
4     d.setTime(d.getTime() + (exdays*12*60*60*1000)); //vrijeme do isteka
5     var expires = "expires="+d.toUTCString();
6     document.cookie = cname + "=" + cvalue + "; " + expires;
7 }
```

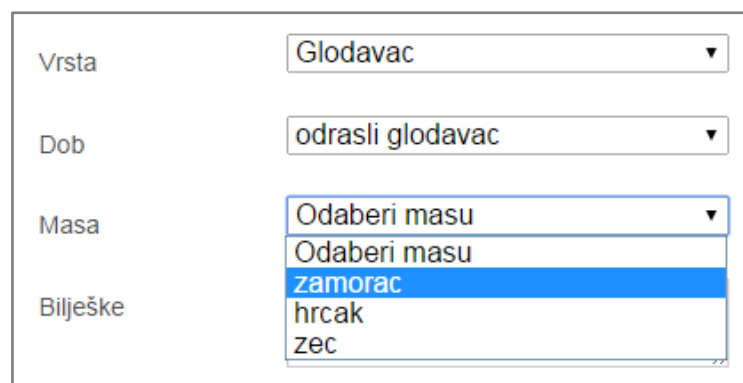
2.8. *Funkcija koju aplikacija APPet koristi za postavljanje kolačića.*

2.3.2. AJAX

AJAX je skraćeni naziv za asinkroni JavaScript u kombinaciji s XML-om. To je skupina međusobno povezanih tehnika web razvoja korištena na klijentskoj strani za izradu asinkronih web aplikacija. Prednost AJAX aplikacija je ta što su neovisne o pregledniku i platformi. Devedesetih godina ovog stoljeća većina web stranica bila je pisana isključivo u HTML-u, a gotovo svaka korisnička akcija zahtijevala je ponovno učitavanje stranice. Takav postupak je neučinkovit i odražava se ne korisničko iskustvo jer usporava učitavanje stranica. Primjena AJAX-a otklanja taj problem. Potrebno je naglasiti da AJAX nije novi programski jezik, već nov način korištenja postojećih standarda. Umjetnost razmjene podataka s poslužiteljem i izmjene

dijelova postojeće stranice u pozadini, bez ponovnog učitavanja. Baš zbog toga, korištenje AJAX-a izvrсна je tehnika za stvaranje brzih i dinamičnih web stranica.

Web aplikacija za vlasnike kućnih ljubimaca koristi AJAX prilikom popunjavanja obrasca s podacima o ljubimcu (Sl.2.9.). Konkretno, AJAX u tom obrascu služi za stvaranje dinamički generiranih padajućih izbornika. Korisnici imaju mogućnost odabrati vrstu svog kućnog ljubimca, a na temelju odabrane vrste, dalje se, uz pomoć AJAX- a, pojavljuju vrijednosti ponuđene u padajućim izbornicima za dob i masu. Sve ponuđene vrijednosti dobivaju se iz baze podataka. Osnovni kocept za dinamički generiran padajući izbornik na temelju odabira u prethodnom izborniku preuzet je s *GitHub* stranice na kojoj se nalazi velik broj skripti otvorenog kôda, a potom prilagođen potrebama aplikacije. Primjer skripte za dinamičko generiranje izbornika korištene u aplikaciji nalazi se na slici 2.9. i odnosi samo na izbornike „Vrsta“ i „Dob“ .



Sl. 2.8. Dio obrasca za upis podataka o ljubimcu koji koristi AJAX.

```
1 <script language="javascript" type="text/javascript">
2 function getXMLHTTP() {
3     var xmlhttp=false;
4     try{
5         xmlhttp=new XMLHttpRequest();
6     }
7     catch(e){
8         try{
9             xmlhttp= new ActiveXObject("Microsoft.XMLHTTP");
10        }
11        catch(e){
12            try{
13                xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
14            }
15            catch(e1){
16                xmlhttp=false;
17            }
18        }
19    }
```

```

19     }
20     return xmlhttp;
21 }
22 function getAge(speciesId) {
23     var strURL="add_pet.php?species="+speciesId;
24     var req = getXMLHTTP();
25     if (req) {
26         req.onreadystatechange = function() {
27             if (req.readyState == 4) {
28                 if (req.status == 200) {
29                     document.getElementById('pet_age').innerHTML=req.responseText;
30                 } else {
31                     alert("Problem while using XMLHTTP:\n" + req.statusText);
32                 }
33             }
34         }
35         req.open("GET", strURL, true);
36         req.send(null);
37     }
38 }
39 </script>

```

Sl. 2.9. *Primjer korištenja AJAX-a u APPet aplikaciji.*

3. POSLUŽITELJSKE TEHNOLOGIJE

Poslužiteljske tehnologije, kako im i samo ime daje naslutiti, obavljaju razne radnje na strani poslužitelja. Njihova je zadaća omogućiti pružanje informacija, odgovoriti na zahtjeve korisnika, izvršiti određene funkcije ili preuzeti podatke iz baze. Ukoliko postoji potreba za dinamičkim web stranicama, za njihovu izradu koriste se programski jezici i baze podataka. Iako postoji nekoliko programskih jezika koji služe toj svrsi, za izradu aplikacije korišten je PHP. Ostali jezici sličnih mogućnosti su *Microsoftov ASP.NET* i *Ruby on Rails*. Osim PHP-a korišten je i MySQL sustav za upravljanje bazama podataka. Za rad s PHP-om i bazama podataka potrebno je instalirati neki od paketa čija je zadaća „pretvoriti“ osobno računalo u web poslužitelj. Za potrebe testiranja i razvoja aplikacije instaliran je paket pod nazivom *WampServer*.

3.1. PHP Hypertext Preprocessor

PHP je skriptni jezik koji se koristi na strani poslužitelja, osmišljen za razvoj web aplikacija, no moguće ga je koristiti i kao opći programski jezik. Kratica PHP primarno je bila kratica za *Personal Home Page*, no danas ona predstavlja izraz *PHP Hypertext Preprocessor*. 2013. godine PHP je pokretao čak 240 milijuna web mjesta. Njegova prednost je što je vrlo fleksibilan i u HTML ga se može umetnuti na različite načine. Korištenjem PHP-a, kontrola web poslužitelja gotovo je neograničena. Bilo da je potrebno izmijeniti HTML, obraditi kreditnu karticu ili dodati podatke o korisniku u bazu podataka, sve se to može obaviti u PHP-u. Dodatan plus je što je slobodan za preuzimanje i uporabu te se može izvoditi na raznim platformama. Primjer složenosti i mogućnosti PHP-a najveća je društvena mreža na svijetu, *Facebook*. Iako zadužen za rad najveće društvene mreže, ipak je dovoljno jednostavan da se smatra „poslužiteljskim jezikom za početnike“. PHP datoteke mogu sadržavati tekst, HTML, CSS, JavaScript i PHP kôdove. PHP kôd se izvodi na poslužitelju, a rezultati se pregledniku vraćaju kao obični HTML.

3.1.1. Primjena i primjeri

Sintaksa PHP-a općenito se smatra vrlo jednostavnom i prikladnom za početnike u programiranju dinamičnih i interaktivnih web mjesta. To proizlazi iz njene sličnosti sa sintaksom programskih jezika C i C++, koji već dugo služe kao školski primjeri za učenje programiranja. PHP kôd piše se unutar PHP oznaka „<?php ... ?>“ i moguće ga je smjestiti bilo gdje u HTML kôdu. Još jedno korisno obilježje PHP-a je postojanje tzv. superglobalnih varijabli. One se

sastoje od nekoliko predefiniраниh varijabli kojima je moguće pristupiti u svakom trenutku, neovisno o dosegu. Može im se pristupiti u funkcijama, klasama i datotekama, bez ikakvih posebnih radnji. Neke od tih varijabli su `$_POST`, `$_SESSION`, `$_SERVER` i dr. Superglobalne varijable korištene su u aplikaciji za obradu podataka iz obrazaca i kreiranje sustava za prijavu i registraciju.

Budući da je cijela *APPet* aplikacija za vlasnike kućnih ljubimaca izrađena u PHP-u, mnogobrojni su načini na koji je ovaj jezik korišten. Prva prednost korištenja skriptnog jezika u izradi aplikacije, u odnosu na korištenje isključivo klijentskih tehnologija, je skraćivanje vremena izrade stranice tako što se smanjuje potreba za kopiranjem određenih dijelova kôda. Primjerice, kroz cijelu aplikaciju proteže se isto HTML zaglavlje u kojem se nalaze poveznice na vanjske stilove, jezične postavke, naslov stranice i sl. Prema tome, jedino rješenje za primjenu tih postavki je kopirati zaglavlje u svaki novi HTML dokument i dodijeliti mu odgovarajući naslov.

Zahvaljujući PHP-u to nije slučaj s *APPetom* jer postoje funkcije *include()* i *require()* za uključivanje vanjskih datoteka. Na taj način je sadržaj zaglavlja potrebno napisati samo jednom i potom ga primijeniti na sve željene stranice. Nedostatak bi u tom slučaju bio što sve stranice imaju identičan naslov, ali ako se umjesto sadržaja naslova ispiše varijabla pomoću funkcije *echo()* i definira u svakom dokumentu koji sadrži zaglavlje, riješen je i taj problem. Isto tako, dodano je i podnožje stranice. Primjer uključivanja vanjskih PHP datoteka vidljiv je u zaglavlju aplikacije (Sl.3.1.). Isti princip primjenjuje se i za uključivanje navigacijske trake. Još jedna primjena PHP-a prikazana na slici 3.1. je aktivnost određene stranice. Primjerice, kada korisnik klikne na karticu „O nama“, ona ostaje istaknuta drugom bojom sve dok je korisnik na toj stranici.

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <title><?php echo $pageTitle;?></title>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6     <link href="<?php echo BASE_URL;?>css/normalize.css" rel="stylesheet">
7     <!-- SVE OSTALE POSTAVKE ZAGLAVLJA -->
8 </head>
9 <body>
10 <!-- NAVIGACIJSKA TRAKA-->
11 <div class="contain-to-grid fixed">
12     <nav class="top-bar data-topbar">
13         <ul class="title-area">
14             <li class="name">
```

```

14         <h1><a href="<?php echo BASE_URL; ?>">APPet</a></h1>
15     </li>
16 </ul>
17     <section class="top-bar-section">
18         <ul class="right"> <!-- PROVJERA AKTIVNOSTI KARTICE-->
19             <li <?php if($section=="about") { echo 'class="active"'; }
20 ?>><a href="<?php echo BASE_URL; ?>about/">O nama</a></li>
21             <li <?php if($section=="contact") { echo 'class="active"';}
22 ?>><a href="<?php echo BASE_URL; ?>contact/">Kontakt</a></li>
23         </ul>
24     </section>
25 </nav>
26 </div>

```

Sl. 3.1. Primjena PHP-a u zaglavlju („header_in.php“) aplikacije.

Primjer na slici 3.2. pokazuje način na koji su zaglavlje *header_in* i podnožje *footer* uključeni u dokument. Ista metoda primjenjena je uvijek kada je potrebno kopirati veći dio kôda nekoliko puta.

```

1 <?php
2 $pageTitle = "O nama"; //naslov stranice pohranjen u varijablu
3 $section = "about";
4 //provjera je li korisnik prijavljen ili ne, ako je, uključuje se
5 //zaglavlje header_in.php
6 include (ROOT_PATH . 'inc/header_in.php'); //ROOT_PATH konstanta
7 ?>                                     //sadrži glavni URL
8 <body>
9     <!--SADRŽAJ STRANICE O NAMA-->
10 <?php
11     //uključivanje podnožja stranice
12     include (ROOT_PATH . 'inc/footer.php');
13 ?>

```

Sl. 3.2. Stranica aplikacije u koju su uključeni zaglavlje i podnožje.

Osim navedenih primjera postoje i brojni drugi primjeri za korištenje PHP-a u sklopu aplikacije. To su *foreach* petlje za izlistavanje niza kao popisa, naredbe za grananje toka *if...else* prilikom provjere prijave i provjere pogrešaka prilikom unosa. Nadalje, naredba *switch...case* upotrijebljena je za kreiranje preporuka, aktivnosti i plana prehrane, a deklarirano je i nekoliko funkcija s različitim namjenama. Kada je riječ o dinamici web aplikacija, značajnu ulogu ima komunikacija s bazom podataka. PHP dohvaća podatke iz baze pomoću *PDO* objekta ili funkcija *mysql_connect()*, *mysql_query()* i dr. Primjer funkcije za dohvaćanje svih ljubimaca iz baze podataka za trenutno prijavljenog korisnika nalazi se na slici 3.3.

```

1 //funkcija koja vraća sve ljubimce pojedinog korisnika
2 function get_pets() {
3     require_once(ROOT_PATH . "inc/database.php");
4     //tries to run a query
5     try {
6         $results = $db->query('SELECT * FROM pets_all WHERE pet_owner="'
7             . $_SESSION['user_id'] . '" COLLATE utf8_bin');
8     } catch (Exception $e) {
9         echo "Pogreška!";
10        exit;
11    }
12    $all_pets = $results ->fetchAll(PDO::FETCH_ASSOC); //podaci u nizu
13    return $all_pets;
14 }

```

Sl. 3.3. PHP funkcija za dobivanje svih ljubimaca prijavljenog vlasnika.

3.2. Baza podataka - MySQL

Nema mnogo smisla dinamički mijenjati HTML izlaz bez praćenja promjena koje korisnici rade dok se služe web stranicom. U ranim danima weba, mnoge su stranice koristile „jednolične“ tekstualne datoteke za pohranu važnih podataka kao što su korisničko ime i lozinka. Takav pristup stvarao je probleme ako direktorij s povjerljivim datotekama nije bio pravilno zaštićen od zloupotrebe. Osim toga, datoteke tog tipa često postaju velike i nespretno za korištenje. Upravo zato su relacijske baze podataka sa strukturalnim upitima postale neizbježne. Tako je i MySQL, sustav za upravljanje bazama podataka, besplatan za korištenje i instalaciju na brojnim internetskim poslužiteljima, postao široko upotrebljavan. MySQL je vrlo velik, ali iznimno brz sustav za upravljanje bazama podataka koji koristi naredbe slične engleskom jeziku, odnosno naredbe SQL (kratica od *Structured Query Language*) jezika.

3.2.1. Primjena i primjeri

Primjena MySQL baze podataka ima za cilj dinamičko generiranje web aplikacija podacima iz baze. To se postiže povezivanjem SQL naredbi s naredbama PHP jezika. Često se vrijednosti iz baze podataka pohranjuju u varijable i nizove te se tako koriste dalje u PHP kôdu. Taj pristup olakšan je tako što PHP ima unaprijed definirane funkcije i objekte za povezivanje s bazom podataka (Sl.3.6.). Jedino što je potrebno jest dobro poznavanje sintakse tih dvaju jezika. Jednostavnost sintakse SQL-a očitovana je i u primjeru sa slike (Sl.3.4.) pomoću kojega je kreirana jedna od ukupno pet tablica u bazi podataka *APPet* aplikacije.

```

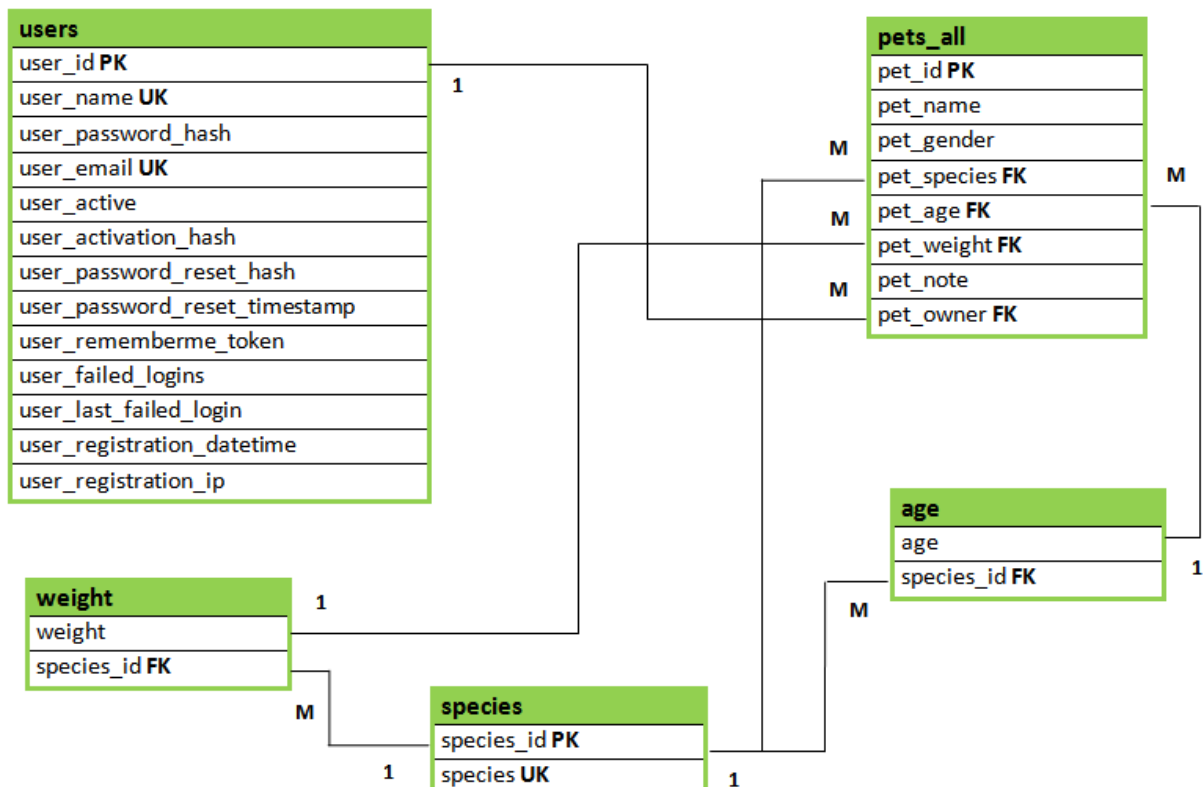
1 CREATE TABLE IF NOT EXISTS `login`.`pets_all` (
2   'pet_id' int(11) NOT NULL AUTO_INCREMENT
3   'pet_name' varchar(50) NOT NULL,
4   'pet_gender' varchar(10) NOT NULL,
5   'pet_species' varchar(20) NOT NULL,
6   'pet_age' varchar(50) NOT NULL,
7   'pet_weight' varchar(50) NOT NULL,
8   'pet_note' text DEFAULT NULL,
9   'pet_owner' int(11) NOT NULL,
10  PRIMARY KEY (`pet_id`),
11  FOREIGN KEY (pet_id) REFERENCES users(user_id),
12  FOREIGN KEY (pet_species) REFERENCES species(species_id),
13  FOREIGN KEY (pet_weight) REFERENCES weight(weight),
14  FOREIGN KEY (pet_age) REFERENCES age(age)
15 ) DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

```

Sl. 3.4. Primjer sintakse SQL jezika za kreiranje tablice „pets_all“.

Cjelokupna baza podataka sastoji se od pet tablica, a sav rad s njom obavljen je u *phpMyAdminu*, besplatnom programskom alatu, namijenjenom za rukovanje MySQL bazom preko Interneta. U *phpMyAdminu* nije čak potrebno pisati SQL kôd, već se baze mogu raditi i preko sučelja, no to nije uobičajen pristup. Prva tablica, *users*, sadrži sve podatke o korisniku. Tablica *pets_all* podatke o svim ljubimcima, a tablice *species*, *weight* i *age* podatke o dostupnim vrstama, dostupnoj dobi za pojedinu vrstu i dostupnoj masi (ili veličini životinje). Tablice čine relacijsku bazu podataka jer su podaci u određenim tablicama međusobno povezani pomoću stranog ključa.

Postoje tri vrste ključeva kad je riječ o bazama podataka. Primarni ključ (engl. *primary key* jedinstveni je atribut u tablici i ne smije biti *NULL*). Primarni ključevi u bazi podataka aplikacije su identifikacijski brojevi korisnika, ljubimaca i vrste. Jedinstveni ključ (engl. *unique key*) je atribut koji naznačuje da svi redovi tog stupaca moraju imati različite vrijednosti kao npr. korisničko ime i adresa elektroničke pošte korisnika. Posljednji, strani ključ (engl. *foreign key*), je atribut koji ukazuje na zavisnost tablice o nekoj drugoj tablici. Strani ključ se koristi za povezivanje tablica, a primjer u bazi podataka aplikacije je taj da se u tablici *pets_all* nalazi stupac s identifikacijskim brojem vlasnika iz tablice *users*. Shematski prikaz baze podataka prikazan je na slici ispod (Sl.3.5.).



Sl. 3.5. Model relacijske baze podataka aplikacije.

Ipak, prilikom izrade aplikacije nije dovoljno samo izraditi tablice u *phpMyAdminu*, već podatke iz njih preuzeti koristeći PHP. Tako se omogućuje željeno generiranje HTML sadržaja podacima iz baze. U primjeru (Sl.3.6.) se nalazi sadržaj datoteka za povezivanje s bazom podataka.

```

1 //povezivanje pomoću PDO-objekta
2 try{
3     $db = new PDO("mysql:host=" . DB_HOST . ";dbname=" .
4         DB_NAME,DB_USER,DB_PASS);
5     $db->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
6     $db->exec("SET NAMES 'utf8'");
7 } catch (Exception $e) {
8     echo "could not connect to the database!";
9     exit;
10 }
11 //povezivanje pomoću funkcije za povezivanje
12 $con = mysql_connect(DB_HOST, DB_USER, DB_PASS);
13 if (!$con) {
14     die('Pogreška: ' . mysql_error());
15 }
16 mysql_select_db(DB_NAME);
  
```

Sl. 3.6. Povezivanje s bazom podataka aplikacije.

4. WEB APLIKACIJA

Aplikacija za vlasnike kućnih ljubimaca sastoji se od tri glavna sloja. Prvi je prezentacijski sloj (engl. *presentation tier*) koji se odnosi na prikaz informacija korisniku web preglednika. Tom sloju pripadaju svi HTML, CSS i JavaScript kôdovi u dokumentima aplikacije. Drugi, aplikacijski sloj (engl. *application tier*) je sloj koji upravlja svim zadaćama koje aplikacija treba obaviti. Aplikacijski sloj *APPeta* ostvaren je pomoću PHP-a. Posljednji, podatkovni sloj (engl. *data tier*), povezan je s upravljanjem bazom podataka aplikacije.

4.1. Prezentacijski sloj

Aplikacija za vlasnike kućnih ljubimaca ima jednostavan, ali moderan dizajn ostvaren pomoću *Foundation 4.0* radnog okvira za CSS. HTML elementima aplikacije dodane su *Foundation* klase koje omogućuju korištenje mreže, gotovih gumba, navigacijske trake i sl. Međutim, aplikacija ima i dodatni CSS dokument (*mystyles.css*) s vlastitim stilovima, klasama i identifikatorima pomoću kojih su prilagođene unaprijed zadane karakteristike. Tako su, na primjer, izmijenjene boje i oblici raznih gumba, promijenjen font, preoblikovani obrasci i slično. Kroz cijelu aplikaciju vidljive su i slike različitih životinja preuzete s Interneta.

Aplikacija je vizualno podijeljena na zaglavlje, podnožje i središnji dio. Budući da se zaglavlje i podnožje protežu kroz sve stranice koje su dio *APPeta*, smješteni su u zaseban direktorij kao zasebni dokumenti te potom, po potrebi, PHP funkcijom *include()* dodani na svaku od stranica. Naravno, zaglavlja su dodana na samom početku HTML dokumenta nakon izvršene provjere o prijavi, a podnožja (Sl.4.3. i Sl. 4.4.) na kraju. Zaglavlje se sastoji od navigacijske trake s karticama koje su poveznice na pojedine dijelove aplikacije. Kako aplikacija ima sustav za prijavu korisnika, navigacijska traka ne izgleda jednako za prijavljene i neprijavljene korisnike. Primjeri navigacijski taka prikazani su na slikama ispod (Sl.4.1. i Sl. 4.2.).








Sl. 4.1. Izgled navigacijske trake za neprijavljene korisnike



Sl. 4.2. Izgled navigacijske trake prijavljenog korisnika „moni“

Središnji dio aplikacije uređen je ovisno o funkciji pojedinog dijela aplikacije. U središnjem dijelu nalazi se npr. popis ljubimaca, podaci o pojedinom ljubimcu, plan prehrane i aktivnosti, informacije o aplikaciji i sl. Neovisno o izgledu središnjeg dijela, i neovisno o tome je li korisnik prijavljen u sustav ili ne, aplikacija ima jedinstveno zaglavlje. U zaglavlju se nalaze poveznice na pojedine dijelove aplikacije, ali samo one kojima mogu pristupiti i prijavljeni i neprijavljeni korisnici. Za moderan izgled zaglavlja korišteni su fontovi u obliku ikonica također preuzeti iz radnog okvira *Foundation*.

 Početna
 O nama
 Kontakt
 Pomoć
 Uvjeti korištenja

Naš glavni zadatak jest usrećiti Vas i Vaše ljubimce. To činimo pronalaskom optimalnog plana prehrane i aktivnosti za vaše najdraže ljubimce.

© 2014 APPet. Sva prava pridržana.

Sl. 4.3. Izgled podnožja aplikacije.

```

1 <div class="row main-footer">
2   <hr class="foot"></hr>
3   <div class="large-4 column text-left">
4     <ul class="link-list">
5       <li><a href="<?php echo BASE_URL;?>"><i class="fi-
6         home"></i>Početna</li>
7       <li><a href="<?php echo BASE_URL;?>about/"><i class="fi-
8         info"></i>Onama</li>
9       <!-- OSTALE KOMPONENTE DODANE NA ISTI NAČIN -->
15    </ul>
16  </div>
17  <div class="large-4 columns text-left copyright push-4">
18    <p>Naš glavni zadatak jest usrećiti Vas i Vaše ljubimce. To činimo
19    pronalaskom optimalnog plana prehrane i aktivnosti za vaše
20    najdraže ljubimce.</p>
21    <p class="copyright">&copy; <?php echo date('Y'); ?> APPet. Sva
22    prava pridržana.</p>
23  </div>
24 </div>

```

Sl. 4.4. Sadržaj dokumenta „*footer.php*“ u kojemu je podnožje aplikacije.

4.2. Aplikacijski i podatkovni sloj

Već je rečeno da aplikacijski i podatkovni sloj čine PHP kôdovi i baza podataka. Zahvaljujući tim dvama slojevima, korisnicima je omogućen pristup različitim funkcijama koje aplikacija pruža. Neke od njih su sustav za registraciju i prijavu, upravljanje popisom ljubimaca, izmjena korisničkih podataka i druge.

4.2.1. Sustav za registraciju i prijavu

Da bi korisnici imali pristup svim funkcijama koje aplikacija pruža, potrebna je njihova prijava u sustav. Sustav prijave i registracije moguće je izraditi uz pomoć PHP programskog jezika korištenjem tzv. superglobalne `$_SESSION` varijable te `session start()` i `session destroy()` funkcija. Kako izrada takvog sustava pripada kategoriji naprednog PHP programiranja, za *APPet* je korišten gotovi sustav prijave i registracije preuzet sa stranice *GitHub.com*. Iako je riječ o gotovoj skripti ipak ju je potrebno prilagoditi zahtjevima aplikacije i pravilno implementirati. Originalna verzija sustava ima mnogo funkcija, ali zadržane su tek neke od njih. Primjerice, registracija korisnika i e-mail aktivacija, prijava i odjava korisnika, sustav za zaboravljenu lozinku te izmjena korisničkih podataka.

Navedene funkcije u potpunosti su prilagođene dizajnu aplikacije, izmijenjen je jezik svih poruka koje javljaju pogrešku ili upozorenje te su neki od korisničkih podataka upotrijebljeni za kreiranje ljubimaca. Originalni sustav prijave i registracije nije razlikovao velika i mala slova što je ostvareno dodavanjem „COLLATE utf8_bin“ u SQL naredbe za pretraživanje baze. Za izgled obrazaca korišteni su HTML elementi „<form>“, „<label>“ i „<fieldset>“, a za prikupljanje podataka, tj. verifikaciju, `$_POST` varijabla kojom se može pristupiti vrijednostima unesenim od strane korisnika i provjeriti jesu li u prihvatljivom obliku, jesu li popunjena sva polja i drugi zahtjevi.

Za pohranu korisničkih podataka sustav koristi MySQL bazu podataka. Radi sigurnosti, lozinke pojedinog korisnika ne pohranjuju se u izvornom obliku već se koristi ugrađena PHP funkcija `password_hash()` kako bi se na temelju posebnog algoritma stvorila nasumična raspršena lozinka (engl. *hashed password*). Da bi prvotna lozinka bila prepoznata prilikom prijave u sustav, koristi se obrnuta funkcija `password_verify()`. Raspršivanje lozinke jedan je od osnovnih sigurnosnih principa ovakvih sustava i minimalna zaštita za svaku stranicu ili aplikaciju na kojoj se koriste lozinke. U slučaju zaboravljene lozinke korisnik treba kliknuti na poveznicu *Zaboravili ste lozinku?* te potom upisati svoje korisničko ime. U slučaju da je ono ispravno, na e-mail adresu korisnika stiže kôd za postavljanje nove lozinke te se nakon toga može ponovno prijaviti u sustav s novom lozinkom,.

Prije učitavanja svake stranice, pomoću jedne od definiranih funkcija sustava za prijavu, provjerava se je li korisnik prijavljen u sustav ili ne (Sl.4.5.). Na temelju ishoda provjere dobiva se informacija o zaglavlju koje je potrebno prikazati. Isto tako i o sadržaju koji je tom korisniku dostupan. U slučaju da neprijavljeni korisnik pokuša upisati URL dijela stranice rezerviranog za

prijavljene korisnike, dobiva poruku upozorenja. Stranice kojima neprijavljeni korisnici mogu pristupiti su početna stranica, stranica za registraciju, stranica s opisom aplikacije, stranica za kontakt i stranice s uvjetima i referencama. Prijavljeni korisnici, s druge strane, imaju mogućnost izraditi profile svojih ljubimaca, dobiti preporuke i mijenjati korisničke podatke.

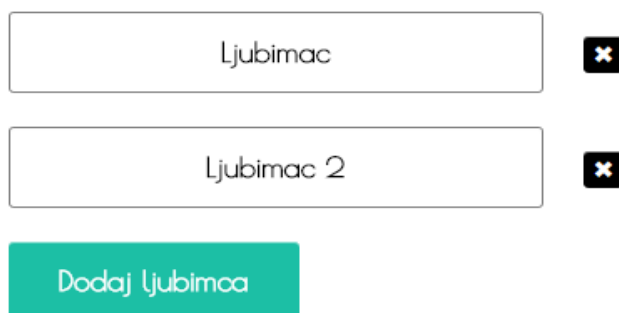
```
1  if ($login->isUserLoggedIn() == true) {  
2      include(ROOT_PATH . 'inc/header_in.php');  
3  } else {  
4      include(ROOT_PATH . 'inc/header_out.php');  
5  }
```

Sl. 4.5. *Kôd koji provjerava je li korisnik prijavljen u sustav.*

Klikom na vlastito korisničko ime, na desnoj strani navigacijske trake, moguće je pristupiti osobnim podacima kao što su *APPet* ime, adresa elektroničke pošte korisnika i broj ljubimaca. Zatim klikom na gumb *Izmjena* korisnik ima mogućnost promijeniti lozinku ili korisnički e-mail. Promjena korisničkog imena nije omogućena jer je ono temelj za identifikaciju korisnika.

4.2.2. Rad s ljubimcima

Prijavljeni korisnici imaju pristup temeljnom dijelu aplikacije za vođenje evidencije o kućnim ljubimcima koje posjeduju. Svaki korisnik aplikacije može dodati željeni broj ljubimaca i za svakog od njih bit ponuđen je plan prehrane i aktivnosti. Nakon prijave u sustav korisniku se otvara sučelje na kojem može vidjeti popis svojih ljubimaca (Sl.4.6.). Popis se mijenja dinamički kako korisnici dodaju i uklanjaju ljubimce i razlikuje se od korisnika do korisnika. Ovaj dio aplikacije komunicira s MySQL bazom iz koje povlači ljubimce kojima identifikacijski broj vlasnika odgovara prijavljenom korisniku. Takav dinamički prikaz ostvaren je pomoću definirane funkcije za prikaz ljubimaca, *foreach* petlje i PHP funkcija za pristupanje bazi podataka. Naredba SQL važna za ostvarivanje ove funkcije je *SELECT* (Sl.3.3.).



The image shows a web form for adding pets. It consists of two text input fields. The first field is labeled 'Ljubimac' and the second is labeled 'Ljubimac 2'. To the right of each input field is a small black square button with a white 'x' icon. Below these two fields is a single green button with the text 'Dodaj ljubimca' in white.

Sl. 4.6. *Isječak iz stranice s popisom ljubimaca.*

Klikom na gumb *Dodaj ljubimca*, otvara se obrazac s podacima koje je potrebno unijeti. U slučaju da neki od njih nedostaje, korisnik dobiva upozorenje. U bazi podataka sadržane su vrste koje aplikacija podržava te podaci o dobi i masi ili skupini, ovisno o određenoj vrsti. Nakon odabira vrste u padajućem izborniku, dinamički se generiraju padajući izbornici za preostala dva područja. Takav sustav dinamički generiranih izbornika ostvaren je pomoću AJAX tehnologije.

Nakon ispunjavanja obrasca, ponovno dolazi do povezivanja s bazom podataka. U tablicu *pets_all* pohranjuju se svi uneseni podaci o ljubimcu, ali isto tako postoji i stupac koji sadrži i identifikacijski broj korisnika kojemu taj ljubimac pripada. Na temelju stupca s podacima o vlasniku, kreiraju se dinamički popisi. Iako je sintaksa vrlo slična za sve funkcije za rad s ljubimcima, ovoga puta ne koristi se SQL naredba *SELECT*, nego naredba *INSERT INTO* za dodavanje novog zapisa u bazu.

U slučaju da je korisnik pogrešno unio podatke o kućnom ljubimcu, omogućena je izmjena nekih podataka. Ipak, ta je funkcija ponajviše dodana zbog dobi ljubimaca koja se mijenja sve dok ne odrastu u potpunosti. Pri tome, korisnik sam treba voditi računa o rastu i mijenjati taj podatak, ukoliko mu je u cilju dobiti pravilne preporuke. Postupak izmjene također se temelji na povezivanju s bazom podataka. Ovoga puta ključ za izmjenu podataka je SQL naredba *UPDATE ()*.

```
1 mysql_query("UPDATE pets_all
2 SET pet_age = '$pet_age',pet_weight = '$pet_weight',pet_note='$pet_note'
3 WHERE pet_id='$pet_id'")
4 //ostatak programa
```

Sl. 4.8. *Naredba za izmjenu podataka o ljubimcu.*

Brisanje ljubimca vrlo je jednostavno i vrši se klikom korisnika na gumb s oznakom križića (Sl.4.6). Kako korisnik ne bi nehotice obrisao ljubimca dodana je JavaScript skripta (Sl.2.6.) koja nudi izbor i traži potvrdu ili odustajanje od brisanja (Sl.2.5. lijevo). U slučaju da je odgovor potvrđan, stranica se ponovno spaja na bazu podataka i korištenjem naredbe *DELETE* briše odabranog ljubimca iz baze koji se više neće nalaziti na popisu ljubimaca.

4.2.3. Preporuke i podsjetnici

Preporuke se sastoje od plana aktivnosti s uputama za igru i vremenom koje ljubimac treba provoditi na otvorenom. Plan prehrane sastoji se od ukupne dnevne količine hrane, općenito preporučenih namirnica i dodatnih savjeta. Podaci povezani s preporukama ne nalaze se

u bazi podataka već u zasebnom dokumentu. Preporuke su temeljene na vrsti i dobi ljubimca, kao i na masi ili tipu. Generiranje preporuka ostvareno je pomoću naredbi za grananje toka programa: *if*, *elseif*, *else* i *switch...case* (Sl.4.9). Preporuke su izrađene uz pomoć web stranica s uputama za skrb o životinjama. Najvažnije od njih su portali *Moj ljubimac*, *Dog Breed Info Center* i web stranica *wikiHow* koja ima veliku bazu raznovrsnih uputa naslovljenih s *Kako se...?*. Ispostavilo se da postoje i brojne detaljne upute za brigu o kućnim ljubimcima raznih vrsta.

```
1  switch ($pet['pet_weight']) {
2      case "do 4 kg":
3          $daily = "Prepouka za hranu 1.";
4          $outside = "Preporuka za šetnju 1.";
5          break;
6      case "4-11 kg":
7          $daily = "Prepouka za hranu 2.";
8          $outside = "Preporuka za šetnju 2.";
9          break;
10     default:
11         $daily = "Preporuka za hranu DEFAULT.";
12         $outside = "Preporuka za šetnju DEFAULT.";
13 }
```

Sl. 4.9. *Primjer kôda za davanje preporuka.*

Aplikacija ima i mogućnost alarmiranja korisnika kako ne bi zaboravili nahraniti svoje ljubimce. Nakon što se korisnik prijavi u sustav i klikne na pojedinog ljubimca, aplikacija će ga pri prvom posjetu tom ljubimcu upozoriti da ga nahrani prema *APPet* uputama. Za podešavanje takvih upozorenja, koja se događaju jednom dnevno ili jednom prilikom učitavanja aplikacije, koristi se JavaScript, točnije kolačići. Kolačići se prema unaprijed definiranim postavkama brišu nakon zatvaranja preglednika, no u aplikaciji im je promijenjen rok trajanja. Osim dnevnog podsjetnika za hranjenje, aplikacija u slučaju da je ljubimac pas, upozorava korisnika kako je pse potrebno cijepiti jednom godišnje. Podsjetnik je ugrađen u stranicu s preporukama i isto tako se ne pojavljuje prilikom svakog otvaranja stranice već traje nekoliko mjeseci. Nedostatak ovakvih kolačića je taj što ih je moguće obrisati u pregledniku, no u ovom slučaju to i nije velik nedostatak jer se podsjetnici pojavljuju periodično i veći broj upozorenja za provjeru preporuka ne može biti na odmet.

4.2.4. Ostali dijelovi aplikacije

Ostali dijelovi aplikacije vidljivi su i prijavljenim i neprijavljenim korisnicima. Prva od njih je stranica za kontakt jer je komunikacija s korisnikom iznimno važna. Tako autori aplikacije mogu dobiti uvid u pogreške koje su previdjeli prilikom izrade, probleme sa funkcionalnošću i sl. Na stranici kontakt nalazi se obrazac koji je potrebno ispuniti podacima i porukom. Sustav za slanje poruka *PHPMailer*, preuzet s *GitHuba*, omogućava slanje elektroničke pošte putem *Gmaila*. Isto tako, poruke s upitima stižu na odabranu adresu, u ovom slučaju na adresu dizajnera aplikacije. Da bi sustav radio i na lokalnom poslužitelju potrebno je obratiti pozornost na dodatne postavke *Apache* poslužitelja .

Ono što je neizmjereno važno kôd ovakvih kontaktnih obrazaca jest sigurnost. Naime, postoji mogućnost da „zloćudni“ strani program šalje tisuće elektroničkih poruka koristeći obrazac aplikacije i time preopteretiti poslužitelje. Iz tog razloga dodano je skriveno polje koje mora ostati prazno kako bi se osiguralo da obrazac ispunjava ljudska osoba, a ne računalni program. Naime, računalni program oslanja se isključivo na izvorni kôd, a ne na izgled te će tako popuniti i to skriveno polje, a u tom slučaju onemogućeno je slanje poruke.

```
1 <form method="post" action= "<?php echo BASE_URL; ?>contact/">
2 <!-- SLIJEDE OSTALA POLJA -->
3   <div class="row hide"> <-- SKRIVENO POLJE -->
4     <div class="large-3 columns">
5       <label for="address" class="left-inline">Address</label>
6     </div>
7     <div class="large-9 columns">
8       <textarea placeholder="Ostaviti prazno!" id="address"
9         name="address"></textarea>
10    </div>
11  </div>
12 <!-- OSTALO -->
```

Sl. 4.10. *Skriveno polje obrasca za kontakt.*

Osim stranice za kontakt postoji i odjeljak *O nama* koji sadrži osnovne podatke o aplikaciji, koja je njena svrha, što se na njoj sve može pronaći i koje su prednosti njenog korištenja. Ovaj dio stranice nije dinamičan kao ostatak aplikacije i naglasak je na privlačnom izgledu. Aplikacija sadrži i dvije sasvim jednostavne stranice *Pomoć* i *Uvjeti korištenja* koje sadrže obavijesti i izvore koji su poslužili za izradu aplikacije.

5. TESTIRANJE

Kao provjera pravilnog rada aplikacije u različitim situacijama odrađeni su testovi značajnijih obrazaca aplikacije. Iako je važno da obrazac funkcionira kada su u njega pravilno uneseni predviđeni podaci, često se događa da korisnik pogrešno unese jedan ili više podataka. Tu nastupaju metode za obradu poslanog obrasca i komunikacija s korisnikom kojemu je važno dati do znanja na što treba pripaziti prilikom sljedećeg unosa. Testiranje je obavljeno ručnim unosom podataka u web pregledniku *Google Chrome* na lokalnom poslužitelju *WampServer 2.5*. Rezultati testiranja pojedinih obrazaca prikazani su u tablicama ispod (Tab.5.1. – 5.7.). Tablice se sastoje od tri stupca. Prvi je redni broj testa, drugi sadrži opise testova, a posljednji poruku aplikacije ili odrađenu radnju.

Tab. 5.1. *Testiranje obrasca za kontakt*

Broj testa	Opis testa	Napomena
1.	Jedno ili više neispunjenih polja	„Potrebno je ispuniti sva polja!“
2.	Sva polja ispunjena, ali polje <i>e-mail</i> nije u zadanom formatu	„Molimo unesite važeću e-mail adresu.“
3.	Sva polja ispravno popunjena	„Poruka je poslana. Hvala!“
4.	Sva polja ispravno popunjena, uključujući i skriveno polje	„Došlo je do pogreške!“

Tab. 5.2. *Testiranje obrasca za prijavu*

Broj testa	Opis testa	Napomena
1.	Sva polja neispunjena.	„Ispunite obrazac.“
2.	Neispunjeno polje za lozinku.	„Niste upisali lozinku.“
3.	Neispunjeno polje za korisničko ime.	„Niste upisali APPet ime.“
4.	Nepostojeće korisničko ime i lozinka.	„Neuspjela prijava.“
5.	Postojeće korisničko ime i pogrešna lozinka.	„Pogrešna lozinka.“
6.	Ispravno korisničko ime, ali 3 puta pogrešno upisana lozinka.	„3 puta pogrešna lozinka. Pričekajte 30s.“
7.	Lozinka postoji u bazi, ali pogrešno korisničko ime.	„Neuspjela prijava.“
8.	Ispravno korisničko ime i lozinka.	Prijava u sustav.

Tab. 5.3. *Testiranje obrasca za registraciju*

Broj testa	Opis testa	Napomena
1.	Neispunjeno jedno ili više polja	„Potrebno je ispuniti sva polja!“
2.	Sva polja uredno popunjena, ali ime ne odgovara zadanom formatu.	"Neodgovarajuće korisničko ime."
3.	Sva polja uredno popunjena, ali e-mail nije u pravilnom formatu.	„Nevažeća e-mail adresa.“
4.	Sva polja uredno popunjena, ali potvrda lozinke nije točna.	„Netočna potvrda lozinke.“
5.	Sva polja uredno popunjena, ali lozinka nije u odgovarajućem formatu.	„Lozinka mora imati najmanje 6 znakova.“
6.	Sva polja uredno popunjena, ali netočno prepisana generirana riječ.	„Pogrešno prepisana riječ sa slike.“
7.	Sva polja uredno popunjena, ali korisničko ime postoji u bazi.	„Zauzeto korisničko ime. Pokušajte ponovo.“
8.	Sva polja uredno popunjena, ali korisnički e-mail postoji u bazi.	"Zauzeto APPet ime ili e-mail adresa."
9.	Svi podaci ispravno uneseni.	Slanje aktivacijskog e-maila.

Tab. 5.4. *Testiranje obrasca za dodavanje ljubimca.*

Broj testa	Opis testa	Napomena
1.	Jedno ili više neispunjenih polja.	„Potrebno je ispuniti sva polja!“
2.	Sva polja ispunjena/odabrana, ali ime ljubimca ne odgovara zadanom formatu.	„Predugo ime!“
3.	Sva polja ispunjena/odabrana, ali bilješka ne odgovara zadanom formatu.	„Preduga bilješka!“
4.	Sva polja ispravno ispunjena/odabrana.	Ljubimac dodan u bazu podataka.

Tab. 5.5. *Testiranje obrasca za izmjenu podataka o ljubimcu.*

Broj testa	Opis testa	Napomena
1.	Jedno ili više neispunjenih polja.	„Potrebno je ispuniti sva polja!“
2.	Sva polja ispunjena/odabrana, ali bilješka ne odgovara zadanom formatu.	„Preduga bilješka!“
3.	Sva polja ispravno ispunjena/odabrana.	Izmjene u bazi podataka.

Tab. 5.6. *Testiranje obrasca za zaboravljenu lozinku.*

Broj testa	Opis testa	Napomena
1.	Polje nije ispunjeno	„Niste upisali APPet ime“
2.	Nepostojeće korisničko ime	„APPet ime ne postoji“
3.	Postojeće korisničko ime	„Poslan e-mail za promjenu lozinke.“
4.	Ispravno ime, ali neuspjelo slanje maila	„Neuspjela izmjena lozinke: ...“
5.	Sva polja ispravno popunjena	„Uspješna promjena lozinke“ i otvaranje obrasca za prijavu
6.	Netočna potvrda lozinke	„Pogrešna potvrda lozinke“ i ponovno otvaranje obrasca za zahtjev za promjenom lozinke.
7.	Sva polja ispravno popunjena	„Uspješna promjena lozinke“ i otvaranje obrasca za prijavu
8.	Lozinka ne zadovoljava zadani format	„Lozinka prekratka (min. 6 znakova)“ i ponovno otvaranje obrasca za zahtjev za promjenom lozinke.

Tab. 5.7. *Testiranje obrasca za izmjenu korisničkih podataka.*

Broj testa	Opis testa	Napomena
1.	Neispunjeno polje za e-mail.	„Niste ispunili potrebno polje.“
2.	Upisan trenutni e-mail.	„E-mail ne može biti isti kao postojeći!“
3.	Upisan e-mail koji postoji u bazi.	„Ovaj e-mail je zauzet.“
4.	Obrazac za e-mail uredno popunjen.	„Uspješna promjena e-mail adrese u: ...“ i izmjena podatka u bazi.
5.	Neispunjeno jedno ili više polja za izmjenu lozinke.	„Potrebno je popuniti sva polja!“
6.	Netočno upisana trenutna lozinka.	„Pogrešno ste upisali staru lozinku.“
7.	Netočna potvrda nove lozinke.	„Potvrda nove lozinke nije valjana!“
8.	Neodgovarajući format lozinke.	„Lozinka mora sadržavati 6 ili više znakova.“
9.	Sva polja uredno popunjena.	„Uspješno ste promijenili staru lozinku. Sljedeći put se prijavite s novom.“

6. ZAKLJUČAK

Web programiranje je iz dana u dan sve traženija i popularnija grana programiranja, no to nikako ne znači da je jednostavno postati web programerom. Razlog tomu je veliki broj tehnologija koje je potrebno spretno kombinirati. Zahvaljujući klijentskim tehnologijama, prezentacijski dio web aplikacije za vlasnike kućnih ljubimaca uspješno je ostvaren. Konačni rezultat je jednostavan, minimalistički dizajn aplikacije s naglaskom na funkcionalnost uz skočne prozore za komunikaciju s korisnikom. Ipak, prikaz aplikacije za mobilne uređaje i tablete, iako je aplikacija responzivna, ostavlja dosta prostora za dodatne prilagodbe. Za to su zaslužni manji, na dodir osjetljivi ekrani, pa su zahtjevi za dobrim korisničkim iskustvom drugačiji nego oni na računalima.

Kad je riječ o aplikacijskom i podatkovnom dijelu, uz pomoć poslužiteljskih tehnologija omogućene su osnovne i najvažnije funkcije za savjetovanje vlasnika o skrbi za ljubimce. Čak su se i rezultati testiranja aplikacije pokazali zadovoljavajućima. Unatoč tomu, mnogo je opcija koje bi još trebalo dodati prije postavljanja aplikacije na javni poslužitelj. Prije svega, veći broj vrsta ljubimaca, bolji sustav upozorenja i poboljšanje sigurnosti podataka. Uzevši u obzir da je aplikacija nastala bez ikakvog predznanja o web tehnologijama, ishod je prilično pozitivan. Ono što je možda najvažnije naglasiti je da se tek nakon izrade aplikacije „otvaraju nova poglavlja“ web programiranja, ponajprije vezana vezana za poboljšanje sigurnosti aplikacija na mreži, tehnike za njihovo brže izvođenje i usuglašenost s različitim vrstama web preglednika.

Literatura

- [1] R.Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, O'Reilly Media, Sebastpool, CA, 2014.
- [2] J.Valade, PHP & MySQL for dummies, Wiley Publishing, New York, 2002
- [3] P.Brođanac, Osnove HTML-a, JavaScripta i PHP-a, Hrvatski savez informatičara, Zagreb, 2002.
- [4] PHP, <http://php.net/docs.php> , kolovoz 2014.
- [5] W3Schools, <http://www.w3schools.com/>, rujan 2014.
- [6] HTML, <http://en.wikipedia.org/wiki/HTML>, rujan 2014.
- [7] Cascading Style Sheets, http://en.wikipedia.org/wiki/Cascading_Style_Sheets, rujan 2014.
- [8] JavaScript, <http://en.wikipedia.org/wiki/JavaScript>, rujan 2014.
- [9] PHP, <http://en.wikipedia.org/wiki/PHP>, rujan 2014.
- [10] Ajax(Programming), [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)), rujan 2014.
- [11] MySQL, <http://en.wikipedia.org/wiki/MySQL>, rujan 2014.
- [12] Web application, http://en.wikipedia.org/wiki/Web_application, rujan 2014.
- [13] Foundation, http://en.wikipedia.org/wiki/Web_application, 2013.

Sažetak

Glavni zadatak je izrada web aplikacije za vlasnike kućnih ljubimaca koja treba imati opcije za prijavu korisnika u sustav, dodavanje ljubimaca te izradu podsjetnika i plana prehrane i aktivnosti na temelju karakteristika koje ljubimac ima. Na takav, praktičan način, potrebno je proučiti ulogu pojedinih web tehnologija u izradi i funkcioniranju aplikacije. Na strani klijenta korišten je opisni jezik HTML, stilski jezik CSS i računalni jezik JavaScript. Pomoću njih ostvaren je moderan dizajn web aplikacije i interakcija s korisnikom putem skočnih prozora. Kad je riječ o tehnologijama na strani poslužitelja, glavnu ulogu u izradi aplikacije imaju besplatni programski jezik PHP i MySQL sustav za upravljanje relacijskim bazama podataka. U tablicama baze podataka pohranjene su sve informacije o korisnicima i ljubimcima, a onda uz PHP korišteni za stvaranje dinamične web stranice. PHP je zaslužan za gotovu svaku funkciju koju aplikacija ima, no za bolje korisničko iskustvo prilikom ispunjavanja obrazaca korištene su i AJAX skripte. Na koncu je izvršeno testiranje svih obrazaca aplikacije.

KLJUČNE RIJEČI: web aplikacija, klijent, poslužitelj, lokalni poslužitelj, HTML, CSS, JavaScript, AJAX, PHP, SQL, relacijska baza podataka

Abstract

The main task of this final thesis is to make a web application for pet owners. Application should have registration and log in system and the ability of adding new pets. It should also generate feeding reminders along with activity and feeding plans based on pets' characteristics. In that way, it is necessary to fully study the part that web technologies take in application development and work. There are few client-side technologies used for developing this web application such as HTML markup language, CSS style sheet language and JavaScript computer programming language. They are all used to make up a modern design and user interaction through pop-up boxes. When it comes to server-side technologies, free PHP programming language and MySQL database management system take the main part in this application. All information about pets and users are stored in database tables and then, with PHP, used to generate content of dynamic web page. PHP is in charge of all available application functions, but for better user experience when filling forms, AJAX scripts are also added. Finally, all application forms were subjected to testing.

KEY WORDS: web application, client, server, localhost, HTML, CSS, JavaScript, AJAX, PHP, SQL, relational database.

Životopis

Monika Čivić rođena je 24.06.1992. godine u Slavonskom Brodu. Osnovnu školu „Mijat Stojanović “ završila je u Babinoj Gredi 2007. godine. Nakon završetka osnovnoškolskog obrazovanja upisuje se u prvi razred II. gimnazije Osijek koju završava 2011. godine. Iste godine upisuje prvu godinu preddiplomskog studija računarstva na Elektrotehničkom fakultetu u Osijeku. Trenutno je student treće godine. Uz studiranje, aktivan je član studentske udruge *IAESTE* koja promiče mobilnost studenata. Upoznata je s korištenjem *Officeovih* paketa, kao i s radom u programskim jezicima *C* , *C++*, web programiranjem i radom s bazama podataka. Tečno se služi engleskim jezikom u govoru i pismu. Služi se i njemačkim jezikom, ali na B1 stupnju. Također, ima položen vozački ispit B kategorije.

Monika Čivić

(Potpis studenta)