Internet aplikacija za evidenciju zalijevanja cvijeća

Aleksi, Sara

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:959179

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-17**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

ZAVRŠNI RAD

INTERNET APLIKACIJA ZA EVIDENCIJU ZALIJEVANJA CVIJEĆA

Sara Aleksi

Osijek, rujan 2016.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

1.2. Poveznica na izrađenu internet stranicu 1

2. ALATI ZA IZRADU INTERNET APLIKACIJE 2

2.1. HTML, CSS i prilagodljivi dizajn 2

2.2. PHP programsko okruženje 4

2.2.1. Automatski pokretane skripte - *Cron Job* 4

2.2.2. Skripta za slanje e-mail poruka - *PHPmailer* 5

2.3. MySQL baza podataka 6

2.4. Java Script i jQuery programsko okruženje 6

2.4.1. Skripta za oblikovanje slika - *cropit.js* 7

2.4.2. Skripta za barkod generator - *jquery-barcode.js* 8

2.4.3. Skripta za prikaz obavijesti - *notify.js* 9

2.4.4. Skripta za odabir datuma - *datepick.js* 10

2.5. Android aplikacija barkod čitača 10

3. IZRADA INTERNET APLIKACIJE 12

3.1. Forme za unos podataka i izgled internet stranice 12

3.2. Struktura MySQL baze podataka 15

3.3. Programska struktura izrađene internet stranice 16

4. ZAKLJUČAK 19

LITERATURA 20

SAŽETAK 21

ABSTRACT 22

ŽIVOTOPIS 23

PRILOG A. IMPLEMENTACIJA MYSQL BAZE PODATAKA 24

PRILOG B. PHP SKRIPTA ZA SLANJE EMAIL PORUKA 25

1. UVOD

Cilj ovog završnog rada je izrada internet aplikacije za evidenciju zalijevanja cvijeća. Namjena aplikacije je podsjetiti vlasnika cvijeta na zalijevanje cvijeta kada je to potrebno. Podsjetnik zalijevanja cvijeća šalje se e-mail porukom svaki dan u određeno vrijeme. Radi lakšeg pristupa informacijama o zalijevanju, svakom cvijetu je potrebno dodijeliti barkod naljepnicu. U aplikaciju treba ugraditi sustav barkod čitača te se podrazumijeva primjena barkod naljepnica. Aplikacija uključuje izradu baze podataka u kojoj se nalaze tablice *cvijeće*, *korisnici* i *zalijevanje*. U tablici *cvijeće* zapisani su podaci o pojedinom cvijetu: naziv cvijeta, slika, period zalijevanja i datum kada je zaliven. U tablici korisnici nalazi se ime, prezime, korisničko ime, lozinka i email adresa. U tablici zalijevanje povezane su prethodne dvije tablice pomoću ID (eng. Identification) oznake korisnika i cvijeta, čime je omogućeno slanje email podsjetnika o potrebnom zalijevanju određenog cvijeća za pojedinog korisnika. Predviđeno je korištenje PHP, HTML, CSS, JavaScript i MySQL alata za izradu opisane internet aplikacije.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je izraditi internet aplikaciju za evidenciju zalijevanja cvijeća. Aplikacija uključuje MySQL bazu cvijeća s nazivom, slikom, informacijama o cvijetu i datumom prethodnog zalijevanja cvijeća. Upozorenje o nezalivenom cvijeću omogućeno je putem e-maila. Svakom cvijetu dodijeljen je barkod s kojim se lakše pristupa podacima određenog cvijeta.

1.2. Poveznica na izrađenu internet stranicu

Izrađena internet stranica dostupna je na slijedećoj privremenoj poveznici.
http://www.etfos.unios.hr/~ssamardzija/

2. ALATI ZA IZRADU INTERNET APLIKACIJE

U ovom poglavlju ukratko su opisani alati koji su korišteni u ovom završnom radu za izradu internet aplikacije za evidenciju zalijevanja cvijeća.

2.1. HTML, CSS i prilagodljivi dizajn

HTML (eng. Hypertext Markup Language) predstavlja standardizirani sustav označavanja teksta unutar tekstualne datoteke kako bi se internet stranice mogle prikazati u različitim oblicima, vrstama i veličinama teksta, raznim bojama, poveznicama i multimedijalnim sadržajima (slika, zvuk, video). Oznake (eng. Tags) su glavna značajka HTML dokumenta. Koriste se za svrstavanje različitih sadržaja u karakteristične oblike prilagođene za prikaz internet stranice [1]. Na slici 2.1. prikazan je primjer HTML zapisa s odgovarajućim prikazom internet stranice.

Slika 2.1. Primjer HTML oblikovanja internet stranice: HTML (lijevo); izgled (desno).

U primjeru na slici 2.1. prikazana je uporaba često korištene oznake za naslov (eng. Heading) <h1>…</h1>, odlomak (eng. Paragraph) … i oblikovanje teksta ….

Primjenom CSS (eng. Cascading Style Sheets Specification) programskog jezika omogućeno je definiranje pravila oblikovanja prikazanog HTML sadržaja. CSS pravila definiraju se oznakom <style>…</style>. Ova pravila govore web pregledniku kako bi sadržaj internet stranice trebao izgledati u smislu: položaja, poravnanja, širine, visine, itd. Primjer CSS oblikovanog sadržaja prikazan je na slici 2.2.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: right;
    color: blue;
}
</style>
</head>
<body>
<h1>CSS oblikovanje naslova</h1>
<h2>CSS oblikovanje podnaslova</h2>
<p>CSS oblikovanje odlomka teksta</p>
</body>
</html>
```

CSS oblikovanje naslova

CSS oblikovanje podnaslova

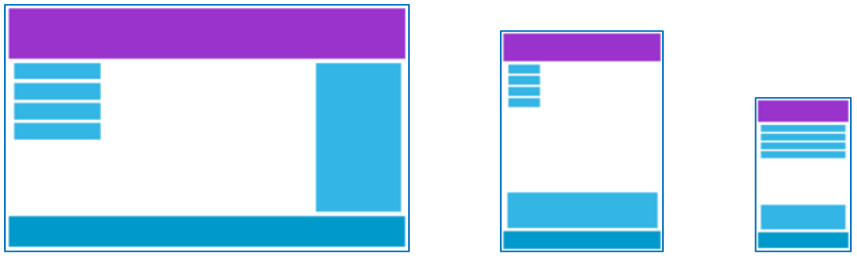
CSS oblikovanje odlomka teksta

Slika 2.2. Primjer CSS oblikovanja internet stranice: program (lijevo); izgled stranice (desno).

U primjeru sa slike 2.2. korišten je standardni HTML opis oznaka sadržaja dok je u CSS datoteci opisano ponašanje sadržaja na stranici. Tekst je poravnat desno i postavljena je plava boja slova. Danas postoje različiti uređaji za prikaz internet stranica s različitim veličinama ekrana. Prilagodljivi dizajn internet stranice (eng. Responsive Web Design) omogućuje prilagodbu HTML sadržaja za određenu veličinu ekrana uređaja. Prilagodljivi (eng. Responsive) dizajn omogućuje se u zaglavlju HTML-a te definicijama oblikovanja u CSS datoteci. Prilagodljivi dizajn internet stranice koristi se za prikaz sadržaja stranice na različitim veličinama ekrana, od velikih do malih prozora preglednika na monitoru, tabletu i mobitelu [5]. Na primjeru u tablici 2.1 prikazan je opis prilagodljivog dizajna, dok je na slici 2.3 prikazan primjer izgleda internet stranice na različitim veličinama.

Tablica 2.1. Primjer HTML i CSS opisa prilagodljivog dizajna internet stranice.

HTML	`<meta name="viewport" content="width=device-width, initial-scale=1">`
CSS	`@media only screen and (max-width:768px) { /* Tableti */ }`
CSS	`@media only screen and (max-width:480px) { /* Mobiteli */ }`



Slika 2.3. Prilagodljivi dizajn internet stranice za različite veličine ekrana. Izvor slike: [5].

Pomoću meta oznake `viewport`, u HTML zapisu je omogućeno upravljanje vidljivim dijelom stranice. Korisnicima je omogućeno mijenjanje širine prozora te skaliranje prikaza stranice. Vrijednost `width=device-width` daje širinu prozora stranice koja se može mijenjati promjenom širine prozora preglednika. Širina prozora stranice je različita za razne uređaje na kojima se stranica prikazuje. S druge strane, u CSS-u je potrebno definirati stilove za različite širine ekrana uređaja. To se postiže primjenom `@media` oznake. U tablici 2.1. je prikazan primjer za tablete i mobitele, dok se za ekrane računala koristi zadani (eng. Default) stil bez `@media` oznake.

2.2. PHP programsko okruženje

PHP je skriptni jezik i prevoditelj (eng. Interpreter) koji je besplatno dostupan i koristi se prije svega na Linux serverima. PHP je učinkovita alternativa konkurentskim alatima kao što je Microsoft ASP. Izvorna kratica PHP dolazi od eng. „Personal Home Page Tools“, a kasnije je postala eng. „Hypertext Preprocessor“ [15].

Glavna značajka PHP programskog okruženja je izvođenje programa na serverskoj strani (eng. Back End), za razliku od Java Script programskog okruženja koji izvršava proigram na klijentskoj strani (eng. Front End). Primjer uporabe PHP programa za stvaranje HTML dokumenata prikazan je na slici 2.4.

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
 <?php echo '<p>Tekst...</p>'; ?>
 </body>
</html>
```

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
  <p>Tekst...</p>
 </body>
</html>
```

Slika 2.4. Uporaba PHP programskog jezika (lijevo) za stvaranje HTML dokumenta (desno).

2.2.1. Automatski pokretane skripte - *Cron Job*

Cron Job je serverska usluga koja omogućuje automatizirano pokretanje određenog programa (PHP skripte) na serveru u unaprijed definirano vrijeme (svaki dan, tjedan, mjesec, …). Vrijeme pokretanja definira se pomoću upravljačke riječi s 5 prisutnih polja. Polja su međusobno odvojena znakom razmaka. Vrijednosti u poljima sadrže vrijeme i datum. Opis ključnih riječi za automatizirano pokretanje skripti je prikazan u tablici 2.2.

Tablica 2.2. Postavke za definiranje vremena pokretanja Cron Job skripte na serveru.

Upravljačka riječ	**"minuta_u_satu sat_u_danu dan_u_mjesecu mjesec_u_godini dan_u_tjednu"**
minuta_u_satu	0 - 59
sat_u_danu	0 - 23, gdje 0 predstavlja ponoć
dan_u_mjesecu	1 – 31
mjesec_u_godini	1 – 12
dan_u_tjednu	0 – 6, gdje 0 predstavlja nedjelju

Pomoću znaka '*' definira se pokretanje u svakom trenutku u određenom polju. Znakom ',' (zarez) odvojene vrijednosti unutar jednog polja omogućuju višestruko pokretanje skripte u određenom vremenskom periodu. Znak '-' omogućuje definiranje vremenskog perioda unutar kojeg se skripta pokreće. Znak '/' omogućuje pokretanje programa „svakog“ određenog vremenskog razmaka. Primjeri ključnih riječi prikazani su u tablici 2.3.

Tablica 2.3. Primjeri ključnih riječi za automatizirano pozivanje Cron Job programa.

Upravljačka riječ	**Vrijeme i datum pokretanja**
`01 04 1 1 1`	u 4:01 sati, 1. dan u mjesecu siječnju te svaki ponedjeljak u siječnju
`01 04 * * *`	u 4:01 sati, svakog dana, svakog mjeseca, svakog dana u tjednu
`01,35 04 * * *`	u 4:01 i 4:35, svakog dana, svakog mjeseca, svakog dana u tjednu
`01 04 1-15 * *`	u 4:01 sati, od 1. do 15. dana, svakog mjeseca, svakog dana u tjednu
`*/10 * * * *`	svakih 10 minuta
`00 06 * * *`	u 6:00 sati, svakog dana, svakog mjeseca, svakog dana u tjednu

2.2.2. Skripta za slanje e-mail poruka - *PHPmailer*

PHPmailer [11] je vrlo poznata i često korištena skripta za slanje email poruka u okruženju programskog jezika PHP verzije 5.0 i novije. Koristi se u mnogim poznatim aplikacijama, među kojima su WordPress i Joomla. Neke od osnovnih karakteristika su slijedeće. PHPmailer podržava SMTP i omogućuje slanje e-mail poruka bez lokalnog mail servera sa SSL i TLS kriptiranjem. Ima mogućnost slanja na više TO, CC, BCC email adresa. Podržava UTF-8 sadržaje i 8-bitno, base64 i binarno kodiranje. Za korištenje je potrebno uključiti *PHPMailerAutoload.php* skriptu koja dohvaća ostale *php* datoteke s klasama za slanje e-mail poruka. Slanje email poruka korisnicima o zalijevanju cvijeća prikazana je u Prilogu B.

2.3. MySQL baza podataka

MySQL (eng. My Structured Query Language) predstavlja otvoren (eng. Open-Source) sustav za upravljanje relacijskim bazama podataka (eng. Relational DataBase Management System). MySQL je popularan odabir baze podataka za primjenu u web aplikacijama te je glavna komponenta LAMP (Lynux, Apache, MySQL, Pearl/PHP) sustava. MySQL se često koristi i u besplatnim (eng. FreeWare) aplikacijama sa sustavom za upravljanje bazom podataka [16].

MySQL se može koristi u kombinaciji s PHP programskim okruženjem. Često korištene naredbe za upravljanje bazom podataka su prikazane u tablici 2.4.

Tablica 2.4. Često korištene naredbe za rad s MySQL bazom podataka.

Naredba	Opis naredbe
`$conn = new mysqli($servername, $username, $password);`	uspostava veze s bazom
`$sql = "SELECT id, firstname, lastname FROM MyGuests";`	upit na bazu
`$result = $conn->query($sql);`	izvršenje upita na bazu
`$row = $result->fetch_assoc();`	dohvat rezultata upita
`mysqli_close($conn);`	prekid veze s bazom

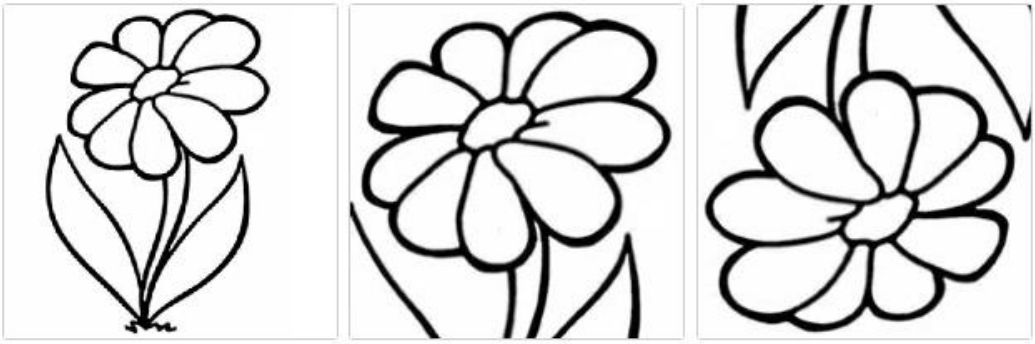
2.4. Java Script i jQuery programsko okruženje

Java Script je programsko okruženje koje se pokreće na klijentskoj strani, a koristi se za upravljanje sadržajem internet stranice u pregledniku. Pomoću Java Script programskog okruženja, sadržaj internet stranice se može mijenjati pozivom funkcija za dohvat HTML elemenata (getElementById, getElementByName), primjenom funkcije za izmjenu vrijednosti (.value) HTML elemenata te funkcije za izmjenu HTML sadržaja (.InnerHTML). Osim izmjene sadržaja, moguće je i upravljanje događajima na stranici. Tako je primjerice moguće upravljati sadržajem stranice pritiskom određene tipke na tipkovnici, klikom miša i sl. Moguće je definiranje i vremenski upravljanih događaja, npr. prikaz trenutnog vremena svake sekunde.

S druge strane, jQuery je brza, malena te značajkama bogata biblioteka funkcija definiranih u Java Script programskom okruženju. jQuery olakšava upravljanje sadržajem HTML dokumenata, rukuje događajima i animacijama. Jednostavno se koristi pomoću API-ja (eng. Aplication User Interface) koji radi na mnoštvu internet preglednika. Kombinacijom svestranosti i fleksibilnosti, jQuery je promijenio način pisanja Java Script programa [1].

2.4.1. Skripta za oblikovanje slika - *cropit.js*

Slike koje se postavljaju na stranicu uobičajeno nisu oblikovane prema unaprijed definiranim karakteristikama jer one nastaju pomoću raznih uređaja u raznim veličinama i orijentacijama. Pojedinu sliku je potrebno prilagoditi radi preglednijeg izgleda internet stranice. U ovom radu primijenjen je cropit.js Java Script [3]. Na slici 2.5 prikazan je postupak obrađivanja slike, prije postavljanja određene slike na internet stranicu.



(a) (b) (c)

Slika 2.5. Učitana slika cvijeta (a), pomaknuta i povećana slika (b), rotirana slika (c).

U primjeru na slici 2.5 prikazana je pravokutna maska za izrezivanje te povećanje/smanjenje učitane slike. Sliku je moguće pozicionirati pomicanjem oko maske te povećati/smanjiti pomoću kliznika, npr. slika 2.5b. Konačan rezultat povećane, rotirane i izrezane slike prikazan je na slici 2.5 c. Uporaba *cropit.js* skripte zahtijeva definiranje veličine prikazane slike, povećanje/smanjenje slike te orijentaciju slike.

Da bi se slika postavila na internet stranicu potrebno je prvo izraditi formu za postavljanje (eng. Upload) slike. Nakon što je slika uređena i izrezana na zadanu veličinu, sliku je moguće prikazati na internet stranici te spremiti u bazu podataka.

Općenito, sliku je moguće pohraniti na server na jedan od dva tipična načina. Prvi način je spremiti sliku u datoteku, a njen naziv i putanju spremiti u bazu podataka. Ovaj način zahtijeva upravljanje putanjama i pohranom datoteka te je složeniji za korištenje. Drugi način je spremiti sliku direktno u bazu kao niz brojeva u base64 formatu [7]. U ovom radu odabran je drugi način spremanja slika u bazu, koji je jednostavniji za korištenje ali uobičajeno zahtijeva 30% više memorije za pohranu slike.

2.4.2. Skripta za barkod generator - *jquery-barcode.js*

U ovom radu korištena je jquery-barcode.js skripta za generiranje EAN-13 barkod oznake [4]. EAN-13 barkod oznaka se sastoji od 12 podatkovnih i jednog zaštitnog znaka. Posljednji 13. znak služi za otkrivanje pogreški, ali ne i njihovo ispravljanje. Primjer EAN-13 zapisa za niz brojeva 001234567890, s izračunatim zaštitnim znakom 5, prikazan je na slici 2.6.

Slika 2.6. EAN-13 barkod niza 001234567890, s izračunatim zaštitnim znakom 5.

EAN-13 barkod označavanje se koristi u cijelom svijetu za označavanje proizvoda. Na slici 2.6 prikazano je 13 znakova, od čega se 12 koristi za označavanje. Skripta *jquery-barcode.js* računa i postavlja zaštitni znak na temelju 12 brojčanih znamenki te generira sliku s linijama barkoda. Zaštitni znak se računa postupkom opisanim u algoritmu 1, [8].

Algoritam 1. Izračun zaštitnog (13.) znaka za EAN-13 barkod oznaku [8].

Ulaz: $x_{12}, x_{11}, x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1$ *// x_1=jedinice, x_2=desetice, x_3=stotice,…*
Izlaz: x_{13}
1: $p = x_2 + x_4 + x_6 + x_8 + x_{10} + x_{12}$ *// zbroj znamenki s parnim mjestima, s desna na lijevo*
2: $n = x_1 + x_3 + x_5 + x_7 + x_9 + x_{11}$ *// zbroj znamenki na neparnim mjestima*
3: $s = (p + 3\,n)$
4: $x_{13} = (10 - s \,\%\, 10) \,\%\, 10$ *// ostatak cjelobrojnog dijeljenja s* 10

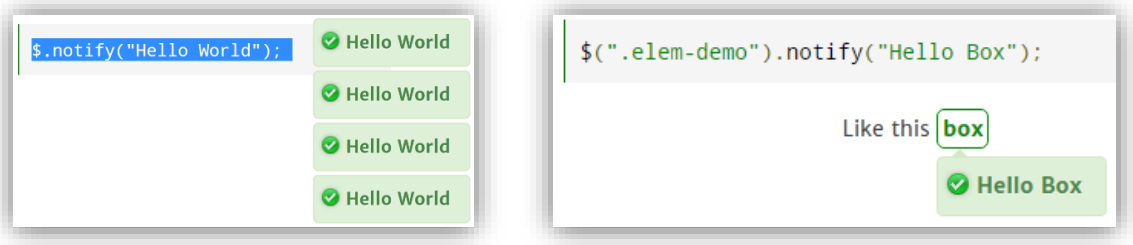
Primjer izračuna sigurnosnog broja $x_{13} = 5$ za broj 001234567890 je slijedeći.

$$p = 9 + 7 + 5 + 3 + 1 + 0 = 25$$
$$n = 0 + 8 + 6 + 4 + 2 + 0 = 20$$
$$s = (25 + 3 \times 20) = 85$$
$$x_{13} = \big(10 - (85\%10)\big)\%10 = (10 - 5)\%10 = 5$$

Prilikom prikaza barkod oznake potrebno je definirati visinu i širinu linija barkoda u točkama (eng. Pixels). Primjer pozivanja funkcije je: barcode("001234567890", "ean13",{barWidth:2, barHeight:30}), a rezultat je prikazan na slici 2.6.

2.4.3. Skripta za prikaz obavijesti - *notify.js*

Notify.js je jQuery dodatak koji pruža relativno jednostavan i potpuno prilagodljiv prikaz kratkotrajnih obavijesti na internet stranici [12]. Vremensko trajanje prikaza obavijesti je podesivo te je početno postavljeno na 5 s. U primjerima na slici 2.7 prikazan je dio programa i rezultat korištenja *apsolutne* obavijesti, koja se odnosi na prozor za prikaz internet stranice, slika 2.7 a, te *relativne* obavijesti koja se odnosi na jedan element unutar internet stranice, slika 2.7 b.



(a) (b)

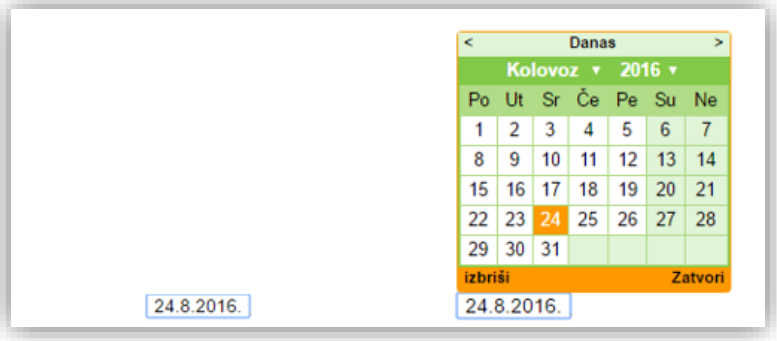
Slika 2.7. Primjer prikaza obavijesti: apsolutni položaj u odnosu na prikaz stranice (a); relativni položaj u odnosu na jedan element na stranici (b).

U primjeru na slici 2.7 korištena je klasa oblikovanja *success* koja prikazuje zeleno obojane obavijesti te označava uspješno obavljenu radnju. Često korištene klase oblikovanja su: *success*, *info*, *warn* i *error*. Primjer pisanja programa za obavijesti s navedenim klasama, kao i njihov rezultat, prikazan je na slici 2.8.

Slika 2.8. Početno ponuđene klase oblikovanja obavijesti (lijevo) te njihov prikaz (desno).

2.4.4. Skripta za odabir datuma - *datepick.js*

Skripta za odabir datuma osigurava pravilan zapis datuma te omogućuje unos datuma u par klikova miša. Skripta *datepick.js* se može postaviti kao *popup* prozor za `<input>` polja ili kao *inline* prozor u `<div>` ili `<span>` poljima s odgovarajućim zadanim postavkama [13]. *Popup* se pojavljuje kada se postavi kursor na `<input>` polje, a nestaje odabirom datuma ili klikom negdje drugdje na stranici. Primjer prikaza sa skrivenom (prije klika) i vidljivom (nakon klika) formom za prikaz datuma prikazan je na slici 2.9.



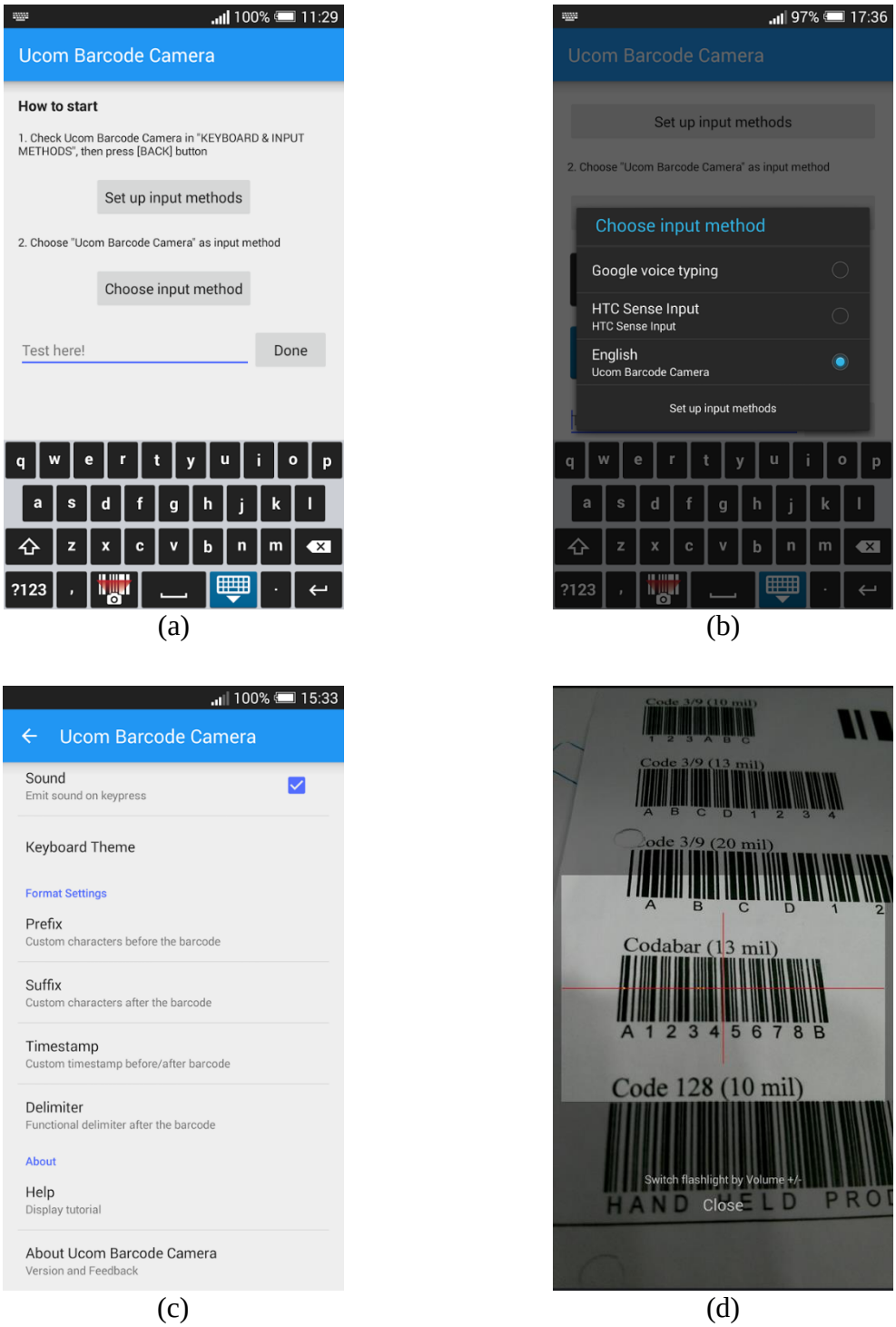
(a) (b)

Slika 2.9. Primjer korištenja *datepick.js* forme za unos datuma: prije klika na `<input>` polje (a); nakon klika na `<input>` polje (b).

2.5. Android aplikacija barkod čitača

Primjenom pametnih telefona s ugrađenom kamerom moguće je pokretanje aplikacije barkod čitača. Barkod čitač obavlja prepoznavanje barkod oznake na temelju slike linija barkoda. Postupak prepoznavanja barkoda nije razmatran u ovom radu. Korištena je gotova android aplikacija koja to obavlja pomoću pametnog telefona s Android sustavom i kamerom. Instalirana je besplatna aplikacija *Ucom Free Barcode Scanner*, koja je preuzeta sa *Google Play Store*-a [9]. Pri prvom korištenju potrebno je omogućiti barkod čitač kao android tipkovnicu, slika 2.10 a. Prilikom korištenja barkod čitača potrebno je odabrati aktivnu android tipkovnicu, slika 2.10 b. Postavke barkod čitača moguće su dugim držanjem na ikonu barkod čitača, nakon čega se pojavi prozor prikazan na slici 2.10 c. Skeniranje barkoda pokreće se klikom na ikonu barkod čitača, koja se nalazi dolje lijevo pored tipke razmaka, na slici 2.10 a. Izgled aplikacije tijekom skeniranja barkoda prikazan je na slici 2.10 d. Nakon očitanja broja, aplikacija se ponaša kao tastatura koja upisuje 13 brojčanih znakova te znak *enter*, koji predstavlja dva ASCII znaka: 10 (LF) (eng. Line Feed) i 13 (CR) (eng. Carridge Return). Nakon upisanog

barkoda, tipka *enter* je potrebna jer se njome obavlja potvrđivanje HTML forme (eng. Submit Form). Unos barkoda pomoću aplikacije ili USB barkod čitača obavlja se kao unos brojčanih znakova s tastature te završava unosom tipke *enter*.



(a) (b) (c) (d)

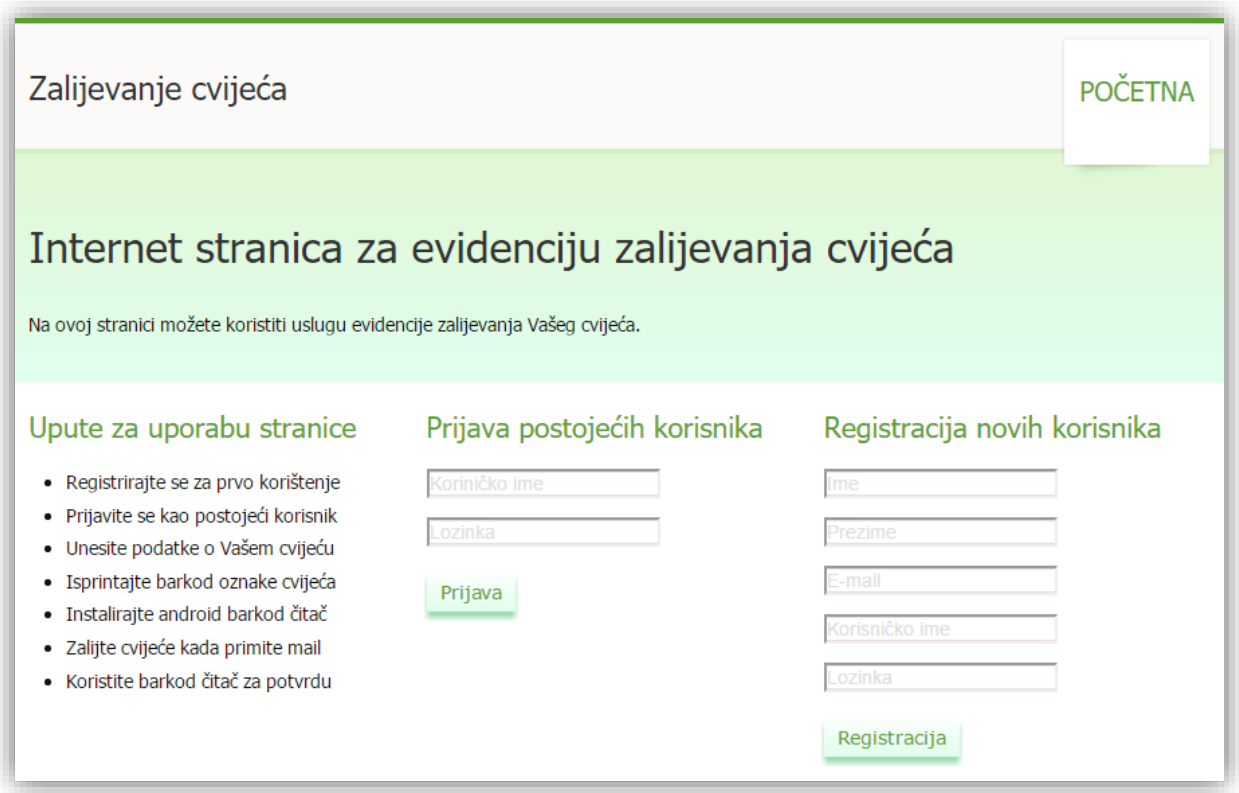
Slika 2.10. Android aplikacija barkod čitača *Ucom Free Barcode Scanner* [9]. Omogućavanje tipkovnice (a); odabir aktivne tipkovnice (b); Postavke barkod čitača (c); Izgled aplikacije (d).

3. IZRADA INTERNET APLIKACIJE

U ovom poglavlju opisani su dijelovi izrađene internet aplikacije. Za izgled stranice iskorišten je CSS predložak (eng. Template) *Rafikov Simpliste Fresh* [14] s unaprijed definiranim klasama oblikovanja HTML elemenata. Primijenjene internet tehnologije su opisane u Poglavlju 2.

3.1. Forme za unos podataka i izgled internet stranice

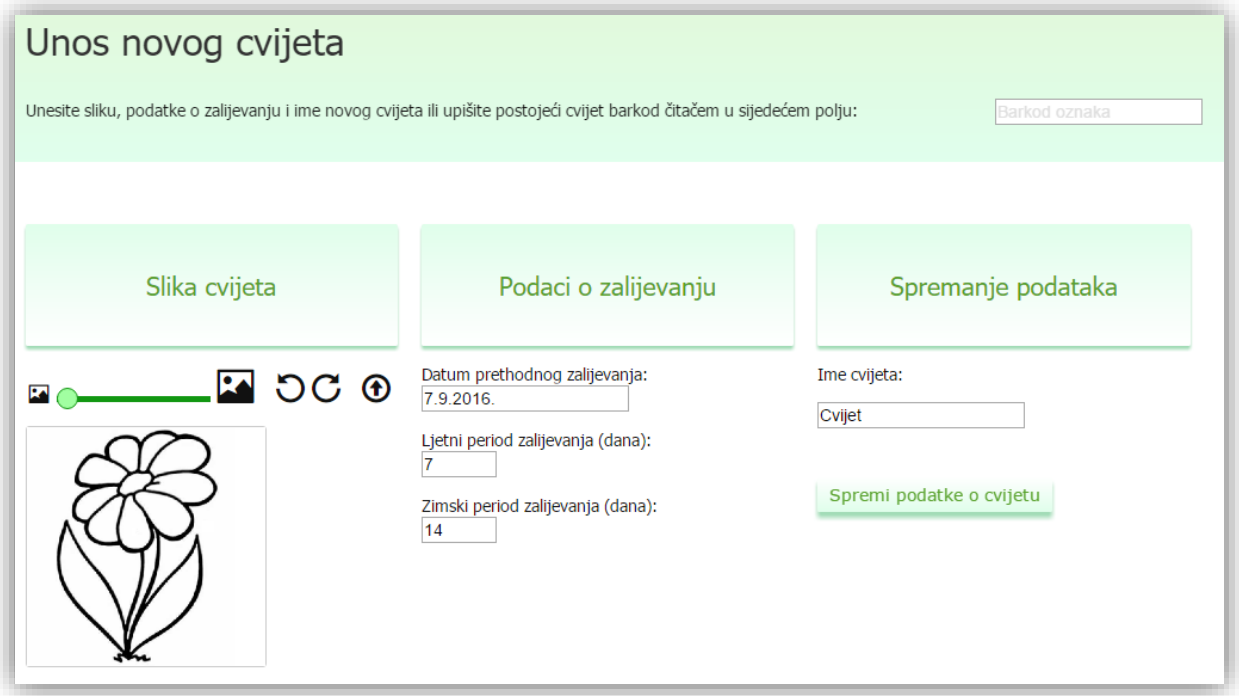
Na početnoj stranici su ponuđene kratke upute za uporabu stranice, postavljene su forme za prijavu postojećih korisnika te registraciju novih korisnika. Nakon što se korisnik prijavi, umjesto forme za registraciju prikazuje se forma za uređivanje korisničkih podataka. Početna stranica, prije prijave/registracije korisnika, prikazana je na slici 3.1.



Slika 3.1. Početna stranica za prijavu i registraciju korisnika.

Nakon prijave/registracije korisnika omogućena je odjava korisnika, uređivanje korisničkih podataka i brisanje korisnika, što je prikazano na slici 3.2. Glavni izbornik je u cijelosti prikazan kada se korisnik prijavi na stranici. Nakon prijave, korisnik bi trebao unijeti podatke o cvijeću. Forma za unos novog cvijeta je prikazana na slici 3.3.

Slika 3.2. Forma za odjavu korisnika i izmjenu osobnih podataka.

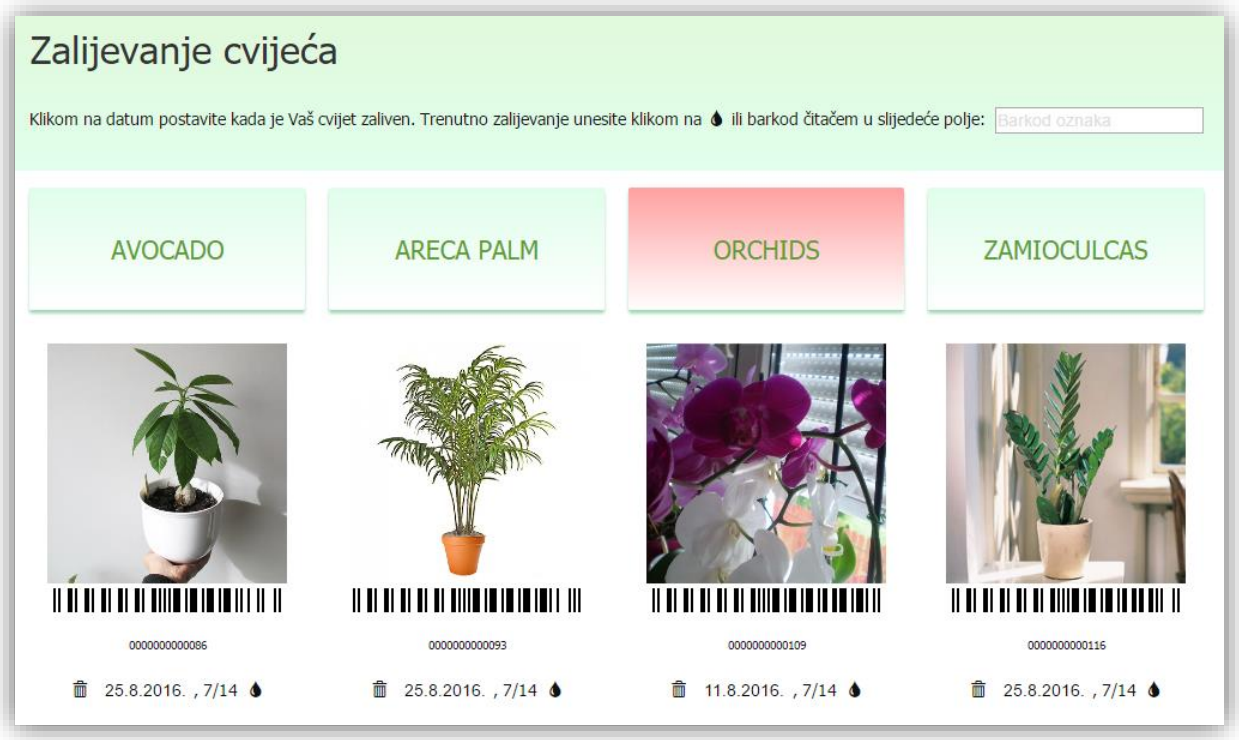


Slika 3.3. Forma za unos novog cvijeta u bazu podataka.

Unos cvijeta moguće je napraviti pomoću barkod čitača u polje „Barkod oznaka“ (gore desno na slici 3.3). Ovim unosom se korisniku pridružuje cvijet koji već postoji u bazi podataka. U ovom radu korišten je EAN13 barkod opisan u poglavlju 2.4.2. Cvijet u bazi ima jedinstvenu ID oznaku koja je iskorištena za izračun barkod oznake. Barkod oznaka dobivena je od brojčane vrijednosti ID cvijeta pretvorbom u niz od 12 ASCII znakova. Primjerice, oznaka ID=1234567890 pretvara se u '001234567890' te mu se dodaje 13. znak '5' koji je izračunat primjenom algoritma 2.1. Time je dobivena jedinstvena barkod oznaka cvijeta '0012345678905'.

Postavljanje slike cvijeta napravljeno je pomoću *cropit.js* skripte [3], koja je opisana u poglavlju 2.4.1. Primjenom ove skripte sve slike cvijeća imaju isto oblikovanje, čime stranica postaje preglednija. Odabir datuma zalijevanja cvijeća obavlja se pomoću *datepicker.js* skripte [13], koja je opisana u poglavlju 2.4.4.

Stranica za zalijevanje cvijeća prikazana je na slici 3.4.



Slika 3.4. Forma za zalijevanje cvijeća.

Na stranici „Zalijevanje“ korisnik može vidjeti cvijeće koje mu je dodijeljeno. Cvijeće koje je potrebno zaliti prikazano je crvenom bojom, dok je ostalo cvijeće prikazano zelenom bojom. Cvijeće je moguće označiti zalivenim na tri načina. Prvi način je klikom na sliku kapljice koja

se nalazi pored slike cvijeta, čime se postavlja današnji datum zalijevanja. Drugi način je unosom barkod oznake u polje „Barkod oznaka“ (gore desno na slici 3.4) čime se postavlja današnji datum zalijevanja. Treći način je klikom na datum zalijevanja te odabirom datuma zalijevanja. Odabir datuma zalijevanja obavlja se pomoću prilagođene *datepick.js* skripte.

3.2. Struktura MySQL baze podataka

U ovom radu korištena je MySQL baza podataka. Baza je strukturirana pomoću tri tablice: *korisnici, cvijeće* i *zalijevanje*. Struktura korištene MySQL baze te nazivi stupaca pojedine tablice prikazani su u tablici 3.1.

Tablica 3.1. Struktura tablica u MySQL bazi podataka.

Naziv tablice	**Nazivi stupaca**					
korisnici	*id*	*user*	*pass*	*email*	*ime*	*prezime*
cvijece	*id*	*naziv*	*slika*	*period_ljeto*	*period_zima*	*datum_zaliven*
zalijevanje	*id*	*id_korisnika*		*id_cvijeta*		

Tablica *korisnici* sadrži podatke o korisničkom računu. Korisnički podaci „user“ i „pass“ su HASH kodirani pomoću funkcije md5(.) radi povećanja razine privatnosti korisnika [6]. Tablica *cvijece* sadrži podatke o pojedinom cvijetu. Polje „slika“ u tablici predstavlja *base64* zapis slike u bazi [7]. Vremenski termin slijedećeg zalijevanja se računa na temelju razlike trenutnog datuma i datuma kada je cvijet zaliven. Ova razlika uspoređuje se s brojem dana aktivnog perioda zalijevanja. Period zalijevanja se uzima kao ljetni period u danima od 1. travnja do 1. listopada, dok je zimski period u danima od 1. listopada do 1. travnja. Tablica *zalijevanje* sadrži ID oznaku korisnika i ID oznaku cvijeta. Pomoću nje je definirano kojem korisniku treba poslati obavijest o tome koji cvijet treba zaliti. Namjena tablice zalijevanje je povezati tablice cvijet i korisnik. Ovime je omogućeno definiranje primanja obavijesti o potrebnom zalijevanju jednog cvijeta prema više korisnika. U tom slučaju, onaj korisnik koji je zalio cvijet treba evidentirati obavljeno zalijevanje cvijeta u bazi.

Osim tablica, u sklopu MySQL baze podataka definirane su *procedure* (eng. Routines) pomoću kojih se u ovom radu obavljaju složeniji upiti na bazu. Korištenjem procedura moguće je definiranje upita na bazu na jednom mjestu uz pozivanje s više mjesta iz više *php* skripti.

Primjerice, procedura *DohvatiCvijeće* se koristi na dva mjesta, prilikom slanja email poruke i prilikom prikaza cvijeća korisnika, a definirana je samo jednom. Jednom definirana procedura izvršava se pomoću SQL naredbe "$CALL\ NazivProcedure(p1, p2, \dots)$", gdje su p_1, p_2, … ulazni parametri procedure. Procedure korištene u ovom radu prikazane su u tablici 3.2.

Tablica 3.2. Procedure u MySQL bazi podataka i njihovi ulazni i izlazni parametri.

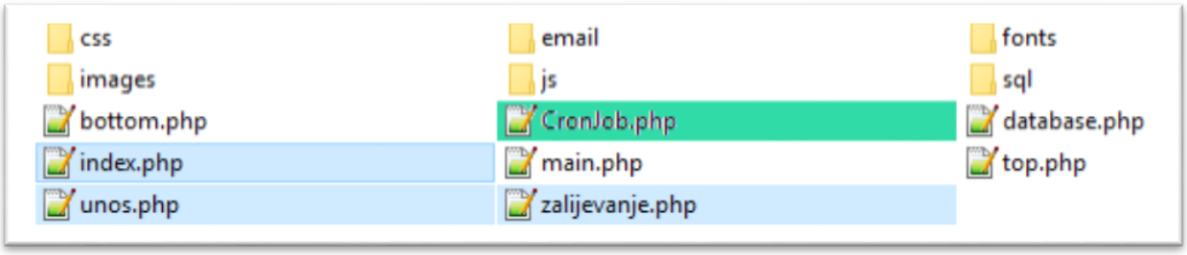
Naziv procedure	**Ulazni parametri**	**Izlazni parametri**
BrisanjeCvijeta	id_cvijeta	/
BrisanjeKorisnika	id_korisnika	/
DohvatiCvijece	id_korisnika	Tablica cvijeća
UnosCvijeta	naziv, slika, period_ljeto, period_zima, datum_zaliven, id_korisnika	/

Procedura *BrisanjeCvijeta* obriše jedan cvijet iz tablica *cvijece* i *zalijevanje*. Procedura *BrisanjeKorisnika* briše jednog korisnika iz tablice *korisnici* te sve unose korisnika u tablici *zalijevanje*. Procedura *DohvatiCvijeće* vraća tablicu s podacima o cvijeću određenog korisnika. Ova procedura najprije dohvaća cvijeće iz tablice *cvijece*, a zatim pomoću uređenog para (*id_korisnika, id_cvijeta*) iz tablice *zalijevanje* uklanja iz rezultata cvijeće koje nije na popisu zalijevanja. Procedura *UnosCvijeta* unosi podatke o cvijeću u tablicu *cvijece* te sprema uređeni par (*id_korisnika, id_cvijeta*) u tablicu *zalijevanje*.

U Prilogu A je prikazan SQL program potreban za stvaranje tablica i procedura u bazi podataka. Navedeni SQL program koristi se za stvaranje baze prilikom postavljanja aplikacije na internet, odnosno pri prvom pokretanju aplikacije. SQL zapis baze podataka daje detaljan uvid u opisane tablice, korištene tipove podataka, ključeve u tablicama i procedure.

3.3. Programska struktura izrađene internet stranice

Na slici 3.5 prikazane su datoteke i mape koje su izrađene za potrebe internet aplikacije opisane u ovom radu.



Slika 3.5. Mape i datoteke izrađene aplikacije.

U tablici 3.3 je opisan sadržaj mapa sa slike 3.5.

Tablica 3.3. Opis sadržaja mapa sa slike 3.5.

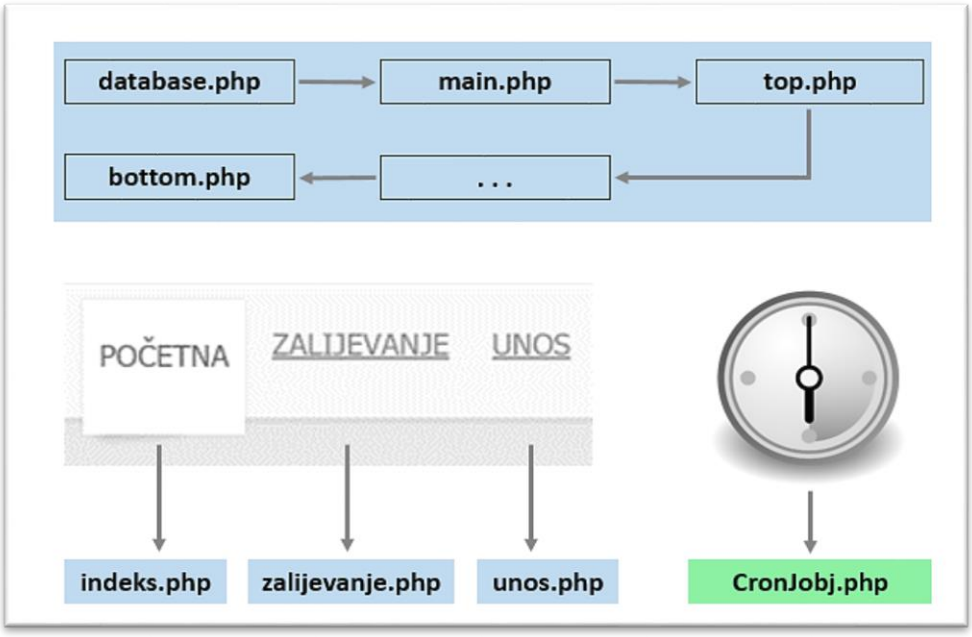
Mapa	**Opis i primjena**
css	Sadrži datoteke za oblikovanje HTML elemenata na internet stranici.
email	Datoteke potrebne za rad *PHPmailer* skripte opisane u poglavlju 2.1.
fonts	Sadrži *glyphicon halflings* font potreban za prikaz za prikaz sličica:
images	Sadrži sliku cvijeta koja se početno postavlja u formu za unos cvijeta, na slici 3.3.
js	Sadrži *jQuery* skriptu te dodatke *datepick*, *cropit*, *notify* i *barcode* opisane u poglavlju 2.4.
sql	Datoteka sa SQL programom za stvaranje baze podataka. Prilog A sadrži ovaj program.

U tablici 3.4 prikazan je popis sadržaja i opis rada pojedine *php* datoteke sa slike 3.5.

Tablica 3.4. Opis pojedine *php* datoteke sa slike 3.5.

Datoteka	**Opis i primjena**
database.php	Uspostavljanje veze s MySQL bazom pomoću korisničkog imena i lozinke.
main.php	Obrada potvrđenih HTML formi metodom post i upravljanje sesijom.
top.php	Ispis početnog dijela HTML dokumenta, ispis glavnog izbornika ako je korisnik prijavljen te uključivanje *java skripti* i *css* oblikovanja u HTML zaglavlju.
indeks.php	Forme za prijavu i registraciju korisnika prikazane na slici 3.1, kao i forme za odjavu, uređivanje podataka korisnika i brisanje korisnika prikazane na slici 3.2.
zalijevanje.php	Forme za zalijevanje cvijeća prikazane na slici 3.4.
unos.php	Forme za unos cvijeća prikazane na slici 3.3.
bottom.php	Ispis copyright podnožja stranice i trenutne godine funkcijom *date*("Y");
CronJob.php	Slanje email obavijesti korisniku za cvijeće koje treba zaliti. Poziva se svaki dan u 6:00 sati pomoću serverske usluge *Cron Job* opisane u poglavlju 2.2.1.

Na slici 3.6 prikazan je redoslijed izvođenja skripti unutar internet aplikacije.



Slika 3.6. Redoslijed izvođenja izrađenih *php* skripti unutar internet aplikacije.

Skripta *CronJob.php* označena zelenom bojom na slici 3.6, slici 3.5 i tablici 3.4, predstavlja automatski pozivanu skriptu svaki dan u 6:00 sati. Skripte označene plavom bojom na slici 3.6, slici 3.5 i tablici 3.4 su glavne tri skripte u izrađenoj internet aplikaciji. Skripte označene bijelom bojom na slici 3.5 i tablici 3.4 se izvode pri izvođenju svake od glavnih skripti, što je prikazano plavom bojom na slici 3.6.

4. ZAKLJUČAK

U ovom završnom radu predstavljena je internet aplikacija za evidenciju zalijevanja cvijeća. U radu su korišteni slijedeći alati za izradu internet aplikacije: PHP, HTML, CSS, JavaScript i MySQL. Primjenom prilagodljivog (eng. Responsive) dizajna omogućeno je pokretanje aplikacije s internet preglednicima pri različitim veličinama ekrana (monitor, tablet, telefon). Izrađena aplikacija podsjeća vlasnika cvijeta na zalijevanje cvijeta slanjem email podsjetnika. Radi lakše evidencije o zalijevanju cvijeća, u aplikaciju je ugrađen barkod sustav označavanja koji je moguće povezati s barkod čitačem te pametnim telefonom. Koristeći ovu aplikaciju vlasnik cvijeća dobiva obavijest o potrebnom zalijevanju cvijeta prema unaprijed definiranom periodu zalijevanja.

LITERATURA

[1] HTML oznake i njihov opis, kolovoz 2016.
http://www.w3schools.com/tags/

[2] Biblioteka JavaScript funkcija – *jQuery*, kolovoz 2016.
https://jquery.com/

[3] Biblioteka za oblikovanje slika – *cropit.js*, kolovoz 2016.
http://scottcheng.github.io/cropit/

[4] Biblioteka za generiranje bar-koda – *jquery-barcode.js*, kolovoz 2016.
http://barcode-coder.com/en/

[5] Prilagodljivi (eng. Responsive) dizajn internet stranice, kolovoz 2016.
http://www.w3schools.com/css/css_rwd_intro.asp

[6] PHP funkcija md5(.) za generiranje HASH koda, kolovoz 2016.
http://php.net/manual/en/function.hash.php

[7] Base64 kodiranje slike, kolovoz 2016.
http://davidbcalhoun.com/2011/when-to-base64-encode-images-and-when-not-to/

[8] Specifikacije EAN-13 barkoda, kolovoz 2016.
http://barcode-coder.com/en/ean-13-specification-102.html

[9] Aplikacija barkod čitača – *Ucom Free Barcode Scanner*, kolovoz 2016.
https://play.google.com/store/apps/details?id=hk.ucom.barcodecamera&hl=en

[10] Automatsko pozivanje PHP skripti – *Croj Job*, kolovoz 2016.
https://help.ubuntu.com/community/CronHowto

[11] PHP skripta za slanje e-mail poruka – *PHPmailer*, kolovoz 2016.
https://github.com/PHPMailer/PHPMailer/blob/master/README.md

[12] Java skripta za prikazivanje kratkih obavijesti – *notify.js*, kolovoz 2016.
https://notifyjs.com/

[13] Java skripta za odabir datuma – *datepick.js*, kolovoz 2016.
http://keith-wood.name/datepick.HTML

[14] „Rafikov Simpliste - Fresh“ CSS template, kolovoz 2016.
http://cssr.ru/simpliste/#fresh

[15] PHP – Hypet Text Preprocesor, rujan 2016.
https://en.wikipedia.org/wiki/PHP

[16] MySQL baza podataka, rujan 2016.
https://en.wikipedia.org/wiki/MySQL

SAŽETAK

Naslov: INTERNET APLIKACIJA ZA EVIDENCIJU ZALIJEVANJA CVIJEĆA

U ovom radu predstavljena je internet aplikacija za evidenciju zalijevanja cvijeća. Namjena aplikacije je pravovremeno podsjetiti vlasnika cvijeta na zalijevanje slanjem email podsjetnika. Pomoću barkod sustava označavanja olakšana je evidencija cvijeća barkod čitačem na pametnom telefonu. Koristeći ovu aplikaciju, vlasnik cvijeća zalijeva cvijet prema unaprijed definiranom periodu zalijevanja, ovisno o vrsti cvijeta te godišnjem dobu.

Ključne riječi: Internet aplikacija, zalijevanje cvijeća, email podsjetnik, barkod označavanje.

ABSTRACT

Title: WEB APPLICATION FOR MONITORED PLANT WATERING

This work addresses internet application for monitored plant watering. The purpose of this application is to remind a plant owner with an e-mail to do the plant watering on time. Efficient plant monitoring is done with barcode labeling system and a smartphone-based barcode reader. Using this application, plant watering is in accordance with a predefined period that vary among plants and the season of the year.

Keywords: Internet application, plant watering, email reminder, barcode labeling.

ŽIVOTOPIS

Sara Aleksi (djevojački Samardžija) rođena je 15.07.1993. g. u Osijeku. Osnovnu školu pohađala je u Petrijevcima. Maturirala je u Općoj gimnaziji u Osijeku 2012. g. Nakon gimnazije upisuje Elektrotehnički Fakultet u Osijeku, preddiplomski stručni studij elektrotehnike, smjer informatika. Priznanje za postignut uspjeh u studiranju dobila je 2015. g. kao najuspješnija studentica smjera informatika. Položila je sve ispite i sada se nalazi pred obranom svog završnog rada.

_____

Sara Aleksi

PRILOG A. IMPLEMENTACIJA MYSQL BAZE PODATAKA

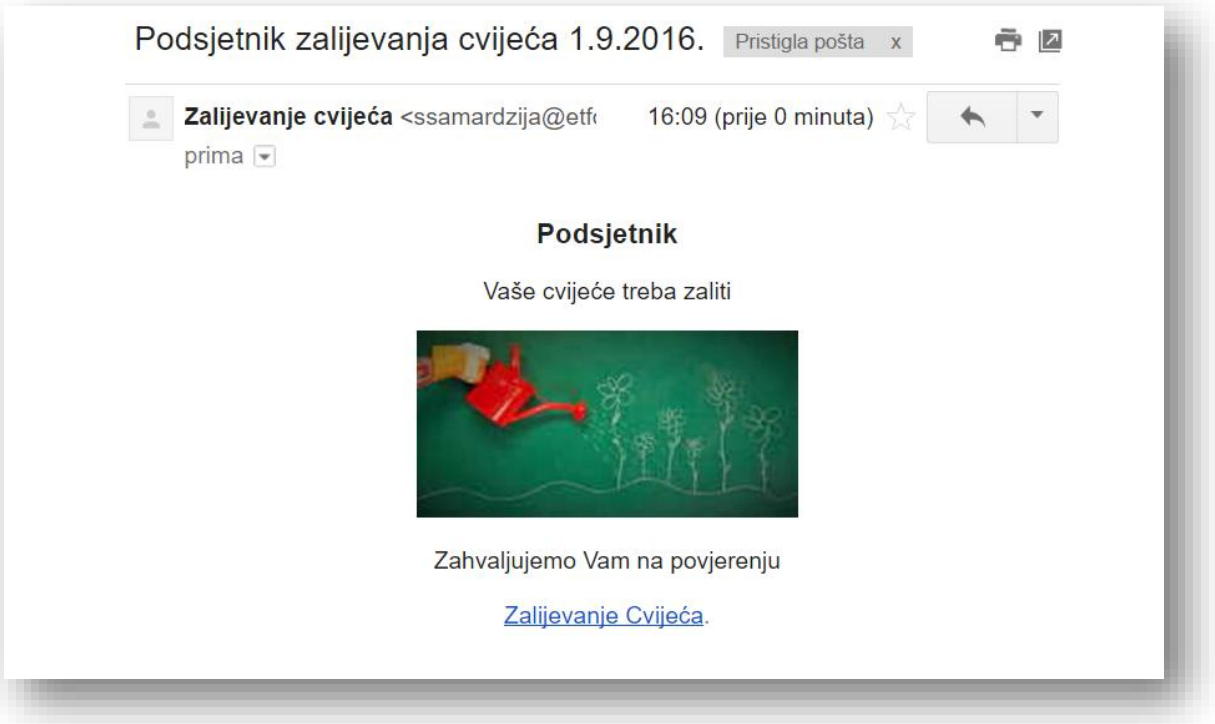
```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

CREATE TABLE IF NOT EXISTS `cvijece` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `ime` text NOT NULL,
  `slika` longtext NOT NULL,
  `period_ljeto` int(11) NOT NULL,
  `period_zima` int(11) NOT NULL,
  `datum_zaliven` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=0;

CREATE TABLE IF NOT EXISTS `korisnici` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user` VARCHAR(32) NOT NULL,
  `pass` VARCHAR(32) NOT NULL,
  `email` text NOT NULL,
  `Ime` text NOT NULL,
  `Prezime` text NOT NULL,
  PRIMARY KEY (`id`),
        UNIQUE KEY `key1` (`user`,`pass`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=0;

CREATE TABLE IF NOT EXISTS `zalijevanje` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_korisnika` int(11) NOT NULL,
  `id_cvijeta` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `key2` (`id_korisnika`,`id_cvijeta`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=0;

DELIMITER //
CREATE PROCEDURE `BrisanjeKorisnika`( IN `user_id` INT(11) )
BEGIN
DELETE FROM `korisnici` WHERE `korisnici`.`id`=`user_id`;
DELETE FROM `zalijevanje` WHERE `zalijevanje`.`id_korisnika`=`user_id`;
COMMIT;
END
//

CREATE PROCEDURE `DohvatiCvijece`( IN `user_id` INT(11) )
BEGIN
SELECT
`cvijece`.`id`,`cvijece`.`ime`,`cvijece`.`slika`,`cvijece`.`period_ljeto`,`cvijece`.`period_zima`,`cvij
ece`.`datum_zaliven` FROM `cvijece` JOIN `zalijevanje` ON `zalijevanje`.`id_cvijeta`=`cvijece`.`id` AND
`zalijevanje`.`id_korisnika`=`user_id`;
END
//

CREATE PROCEDURE `UnosCvijeta`( IN `i` TEXT, IN `s` LONGTEXT, IN `p1` INT(11), IN `p2` INT(11), IN `d`
TEXT, IN `user_id` INT(11) )
BEGIN
INSERT INTO `cvijece`(ime,slika,period_ljeto,period_zima,datum_zaliven) VALUES (`i`,`s`,`p1`,`p2`,`d`);
SELECT LAST_INSERT_ID() INTO @cvijet_id;
INSERT INTO zalijevanje VALUES (NULL,`user_id`,@cvijet_id);
COMMIT;
END
//

CREATE PROCEDURE `BrisanjeCvijeta`(IN `flower_id` INT(11))
BEGIN
DELETE FROM `cvijece` WHERE `cvijece`.`id` = `flower_id`;
DELETE FROM `zalijevanje` WHERE `zalijevanje`.`id_cvijeta` = `flower_id`;
COMMIT;
END
//
```

PRILOG B. PHP SKRIPTA ZA SLANJE EMAIL PORUKA

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
</head>
<body>
<?php
   require 'email/PHPMailerAutoload.php';
   $mail = new PHPMailer();

   // postavke kriptiranog slanja email poruke
   $mail->IsSMTP(true);
   $mail->Host = "drava.etfos.hr";
   $mail->SMTPAuth = true;
   $mail->Username = 'ssamardzija@etfos.hr';
   $mail->Password = '*********';

   $mail->From="ssamardzija@etfos.hr";
   $mail->FromName="Zalijevanje cvijeća";
   $mail->Sender="ssamardzija@etfos.hr";
   $mail->IsHTML(true);
   $mail->CharSet="utf-8";

   $mail->Subject = "Podsjetnik zalijevanja cvijeća " . date("j.n.Y.");

   //slika logo.jpg, oznaka logo
   $mail->AddEmbeddedImage('images/logo.jpg',
                  'logo','images/logo.jpg');
   $mail->Body = "<h3 align='center'>Podsjetnik</h3>
      <p align='center'>Vaše cvijeće treba zaliti</p>
      <p align='center'><a href='http://www.etfos.unios.hr/~ssamardzija/zalijevanje.php'>
         <img src='cid:logo'/></a><br></p>
      <p align='center'>Zahvaljujemo Vam na povjerenju<br>
         <br><a href='http://www.etfos.unios.hr/~ssamardzija'>Zalijevanje Cvijeća</a>.</p>";

   // povezivanje s bazom
   include "database.php";

   // petlja za svakog korisnika
   $q = "SELECT * FROM korisnici";
   $rUsers = mysqli_query($con, $q);            // query result for users
   $nUsers = mysqli_num_rows($rUsers);          // number of users

   while($nUsers--)
   {
      $brCvijeca = 0; // broj cvijeća za zalijevanje

      $user = mysqli_fetch_row($rUsers); // dohvati slijedećeg korisnika

      $q="CALL DohvatiCvijece({$user[0]})";
      $rFlowers = mysqli_query($con, $q);
      $nFlowers = mysqli_num_rows($rFlowers); // broj cvijeća od korisnika

      while($nFlowers--)
      {
         $flower = mysqli_fetch_row($rFlowers); // dohvati slijedeći cvijet

         $p1=$flower[3]; // period ljetni
         $p2=$flower[4]; // period zimski
         $d1=$flower[5]; // datum zaliven

         list($d,$m,$y)=explode('.',$d1); // 5.10.2016. -> 5, 10 i 2016
         $zaliven = mktime(0,0,0,$m,$d,$y); // datum zaliven
         $danas   = mktime(0,0,0,date("m"),date("d"),date("Y"));//današnji

         $d = floor(($danas-$zaliven)/86400); // razlika datuma u danima

         if($d>=0){
            $month = date("m");
```

```
            $d = ($month>3 && $month<10) ? $d-$p1 : $d-$p2;
            // ljetni(4,5,6,7,8,9) : zimski(10,11,12,1,2,3)
            if($d>=0) $brCvijeca++; // broj cvijeća za zalijevanje
         }
      }
      echo "<p>{$user[4]} {$user[5]} treba zaliti $brCvijeca cvijeća</p>";
      if($brCvijeca>0) // ako ima cvijeća za zalijevanje
         $mail->AddBCC($user[3],"{$user[4]} {$user[5]}"); // mail na popis
}

// pošalji podsjetnike
if(!$mail->Send())
   echo "<h3>Pogreška pri slanju: " . $mail->ErrorInfo . "</h3>";
else echo "<h3>Podsjetnici su poslani</h3>";
?>

</body>
</html>
```

Kao rezultat gore navedene *php* skripte stvoren je e-mail podsjetnik prikazan na slici B.1.



Slika B.1. Izgled e-mail podsjetnika o potrebnom zalijevanju cvijeća.