

Web aplikacija za pomoć u praćenju rada vinogradarskog gospodarstva

Krušarovski, Mladen

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://um.nsk.hr/um:nbn:hr:200:129974>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-27**

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
Stručni studij

Web aplikacija za pomoć u praćenju rada
vinogradarskog gospodarstva
Završni rad

Mladen Krušarovski

Osijek, 2016.g.

1. UVOD	1
2. HTML i CSS	2
3. SQL BAZA PODATAKA	5
3.1. Opis pojedinih modela.....	5
3.2. Baza podataka za web aplikaciju	7
3.2.1. Izrada baze podataka	7
3.3. Prikaz strukture baze podataka	11
3.4. SQL naredbe za izradu i popunjavanje baze podataka	11
4. PHP skriptni jezik.....	13
4.1. Projekt.....	14
4.2. Osposobljavanje web aplikacije	14
5. Web aplikacija.....	15
5.1. Login i unos podataka u bazu.....	17
6. Zaključak	24
Literatura	25
Sažetak	26
Abstract	27
Životopis	28

1. UVOD

Ovaj projekt je zamišljen kao funkcionalna web aplikacija koja bi služila u svrhu evidentiranja sezonskih radova u vinogradarskom gospodarstvu. Sama aplikacija izrađena je u svrhu učenja web tehnologija te je izrađena u sklopu završnog rada. Aplikacija bi trebala biti splet različitih tehnologija koje će se pomno opisati unutar svakog poglavlja koje će se sastojati od kratkog teorijskog dijela, odnosno uvoda, te od praktičnog dijela, odnosno detaljne izrade i prikaza izgleda i dizajna, web aplikacije.

Nakon kratkog uvoda slijedi prvo poglavlje koje će prikazati i opisati detaljnu izradu web aplikacije, odnosno stilske dijela koji obuhvaća HTML te CSS stilske jezika. Nakon stilskih jezika slijedi opis baze podataka i njoj pripadajućih komponenti, odnosno jezik SQL koji služi za izradu baze podataka i rad s istom. Predzadnje poglavlje prikazuje PHP skriptnog jezika.

Tehnologije koje će se koristiti pri izradi same aplikacije su: HTML i CSS kao stilski jezik, PHP kao skriptni jezik koji će povezivati bazu podataka napravljenu uz pomoć SQL jezika sa samom web aplikacijom.

2. HTML i CSS

HTML je kratica za *HyperText Markup Language* te je on jezik za označavanje kojim se opisuje web stranica. Najveći razlog opće prihvaćenosti i popularnosti HTML jezika je njegova jednostavna uporaba i brz proces učenja, a daleko najvažniji čimbenik je zamisao da od početka bude besplatan i dostupan svima. Cilj HTML jezika je zapravo uputa web pregledniku kako prikazati hipertekst dokument, što znači da sam web preglednik omogućava prikaz hipertekst dokumenta te se pri tome nastoji da svaki određeni dokument izgleda jednako unutar bilo kojeg web preglednika i bilo kojeg operacijskog sustava.

Dovoljan je i najobičniji uređivač teksta Notepad za izradu stranice pomoću HTML-a. Danas postoji veliki broj raznih programa koji se mogu besplatno preuzeti s Interneta te koji dosta olakšavaju samu izradu web stranica. Za razliku od Notepada, koji zahtijeva ručno uređivanje teksta, ti programi imaju i vizualno uređivanje koje ima prednosti kao što su vidljivost sadržaja web stranice odmah nakon napisanog koda ili mogu sami generirati potreban kod. Na taj se način smanjuje mogućnost grešaka u pisanju koda.

HTML nije programski jezik niti se njime može izvršiti ikakva zadaća ili operacija, on je opisni jezik kojim se opisuje hipertekst dokument. Dokumenti takvoga tipa su zapravo najobičnije tekstualne datoteke samo što imaju posebnu ekstenziju *.html* ili *.htm*, i građene su uz pomoć „tagova“ odnosno oznaka koje zadaju sadržaj koji će biti prikazan unutar web preglednika. [1].

HTML jezik određen je znakovima te svaki početni znak koji otvara polje određenog unosa mora imati i kraj. Na primjer, pri početku pisanja HTML dokumenta ili takozvanog hipertekst dokumenta potrebno je navesti element `<html>` kako bi sam dokument prihvatio da se radi o stilskom jeziku odnosno HTML dokumentu te je potrebno na kraju navođenje oznake `</html>` što Internet pregledniku govori da je kraj HTML dokumenta.

Svi HTML dokumenti se sastoje od dva dijela: zaglavlja (engl. head) i tijela (engl. body). Zaglavlje se odvaja tagovima `<head>` i `</head>`, a tijelo dokumenta tagovima: `<body>` i `</body>`. Sve ono što se napiše u zaglavlju dokumenta neće se prikazati u prozoru Internet preglednika, već koristi samo za pružanje nekih informacija o stranici. [2].

Isto tako se unutar hipertekst dokumenta pozivaju CSS pravila, odnosno određene slike koje predstavljaju „gumb“ ili bilo kakav drugi element, a kao i CSS jezik moguće je pozvati i

ostale jezike kao npr. PHP koji se pri pozivanju unutar nekog bloka označava sa znakovima „<?“. Ali isto tako kako postoji pozivanje pojedinih elemenata unutar html dokumenta, postoje i ograničenja koja su se pokušavala otkloniti daljnjim nadogradnjama HTML jezika. Svaka nadogradnja donosila je pojedine novosti i mogućnosti, a HTML 5 je zamišljeno kao jezik koji će se stalno nadograđivati.

CSS je kratica za Cascading Style Sheets. To je stilski jezik koji služi za oblikovanje i prezentaciju, odnosno za definiranje izgleda dokumenta napisanog u HTML jeziku.

Prvenstvena namjena CSS-a je, uvjetno rečeno, "rasterećenje" HTML-a od definiranja načina prikazivanja sadržaja. Konkretno, pomoću CSS stilskog jezika definira se izgled i raspored sadržaja web stranice. Dakle, pomoću CSS-a se može definirati, između ostalog, izgled i veličinu teksta (font) te boje slova, izgled odnosno veličinu slike. [3].

Na taj način poboljšava se pristup samom sadržaju i dobija se veća fleksibilnost te praktički apsolutna kontrola u specifikacijama prezentacije sadržaja. Prednost CSS-a je mogućnost "pozivanja" različitih stilova za pojedine elemente web stranice. Radi se o takozvanim pravilima preko kojih CSS određuje prioritete i kreira "shemu" za stilove i pravila.

Trenutačno postoje tri inačice CSS-a; CSS 1, CSS 2 i CSS 3. Treba naglasiti da je CSS 2 najviše upotrebljavan, CSS 1 polako odlazi u povijest dok je CSS 3 trenutačno u fazi svojevrsnog testiranja. CSS 2 je još 1998. godine definiran kao jedan od standarda i objavljen je kao preporuka. Dakle, CSS-om se mogu definirati razni elementi web stranice, odnosno prikaz njezinog sadržaja, izgled, stil i veličina slova, boja teksta, pozadine, tekstualni atributi (razmak između slova, riječi ili redaka), poravnanje teksta, slike, tablice, margine, granice, padding, pozicioniranje, jedinstvena identifikacija i općenito klasifikacija grupe atributa[1][3].

U današnje vrijeme razvoja web aplikacija i Internet stranica nemoguće je zamisliti profesionalnu web stranicu koja u sebi nema CSS. Preporuka je da se definicije elemenata sadržaja određuju upravo kroz CSS, u suprotnom stranica neće proći ovjeru niti HTML, a niti, naravno, CSS. Kako bi se moglo objasniti ukratko što, zapravo, "radi" CSS, bit će navedeni jednostavni primjeri. CSS tako možemo "pisati" unutar same HTML stranice na dva načina.

Prvo, kao stilove u zaglavlju dokumenta što prikazuje sljedeći programski odsječak:

```
<style type="text/css">
    h1 { color: red }
</style>
```

Programski odsječak 2.1. Zadavanje CSS-a u zaglavlje HTML dokumenta

Na ovaj način dobila se definicija stila za H1 formatiranje teksta, odnosno boju tog teksta. Iako se sve ovo može, naravno, definirati kroz HTML, preporuka je i standard da se to definira upravo kroz CSS. CSS može se koristiti i unutar samih HTML tagova kako bismo definirali, primjerice, boju nekakvog teksta. Ako se želi dobiti tekst crvene boje njega treba definirati kroz CSS na sljedeći način:

```
<p style="color: red;">Crveni tekst</p> [3].
```

Programski odsječak 2.2. Postavljanje boje teksta

3. SQL BAZA PODATAKA

Baza podataka je organizirana zbirka podataka. Termin je izvorno nastao unutar računalne industrije. Svaki se zapis za bolji povratak i razvrstavanje obično prepoznaje kao skup elemenata (činjenica) podataka. Predmeti vraćeni u odgovoru na upite postaju informacije koje se mogu koristiti za stvaranje odluka koje bi inače mogle biti mnogo teže ili nemoguće za stvaranje. Računalni program korišten za upravljanje i ispitivanje baze podataka nazvan je sustav upravljanja bazom podataka (SUBP). Svojstva i dizajn sustava baze podataka uključeni su u proučavanje informatičke znanosti.

Središnji koncept baze podataka jednak je onome od zbirke zapisa ili dijelova znanja. Za danu bazu podataka tipično postoji strukturni opis vrste činjenica sadržanih u toj bazi podataka: taj opis naziva se shema. Shema opisuje predmete koji su prikazani u bazi podataka te odnose među njima. Postoje brojni različiti načini organiziranja sheme, tj. modeliranja strukture baze podataka: oni se zovu modeli baza podataka (ili modeli podataka). Model u najraširenijoj upotrebi danas je odnosni model koji, laički rečeno, prikazuje sve informacije u obliku mnogostrukih odnosnih tablica od kojih se svaka sastoji od redova i stupaca (prava definicija koristi matematičku terminologiju). Ovaj model prikazuje odnose uporabom vrijednosti koje su zajedničke za više od jedne tablice. Ostali modeli poput hijerarhijskog modela i mrežnog modela koriste prikaze i odnose koji su mnogo eksplicitniji[4].

3.1. Opis pojedinih modela

Hijerarhijski model prilično je jednostavan i ne zahtjeva veliko informatičko znanje, ali su pristup i manipulacija informacijama iznimno otežani zbog stroge hijerarhijske međuovisnosti. Mrežni je model fleksibilan, moćan i efikasan, a podacima se pristupa preko mreže logičkih veza. Nedostatak ovog pristupa jest zahtjev za upotrebom brzih računala i većim informatičkim znanjem.

Relacijski model (RDBMS) temelji se na klasičnom tabličnom modelu. Tablice se međusobno povezuju relacijama čija je glavna uloga čuvanje integriteta podataka, brže pretraživanje i izbjegavanje redundancije odnosno ponavljanja pojedinog podataka na više

mjesta ili tablica. Tablice se povezuju preko stupaca koji se nalaze u obje tablice, a važno je da je riječ o podatku čija je jedinstvenost osigurana. To znači da dva zapisa ne mogu imati istu vrijednost stupca pa se najčešće koriste različite oznake. U „roditeljskoj“ tablici (parent) takav se stupac naziva primarni ključ, a u podređenoj tablici (child), odnosno onoj s kojom je roditeljska (parent) tablica povezana preko primarnog ključa, strani ključ. Ponekad se jedinstvenost može osigurati samo kombinacijom dvaju ili više polja, a takav ključ zove se složeni ključ. Prednost relacijskog modela leži u povezanosti tablica preko ključeva. Primjer: brisanjem člana neke knjižnice u tablici s podacima o članovima automatski možemo izbrisati i sve zapise o njegovim dotadašnjim posudbama u nekoj drugoj tablici. Na taj se način postiže referencijalni integritet odnosno postojanost svih podatak u bazi[5].

SQL (Structured Query Language) najpopularniji je računalni jezik za izradu, traženje, ažuriranje i brisanje podataka iz relacijskog modela, a razvio ga je IBM početkom 70-ih godina. American National Standard Institute (ANSI) i International Standards Organization (ISO) standardizirali su SQL 1986. godine (verzija SQL-86). Trenutno važeći standard datira iz 1992. odakle potječe i novi naziv SQL-92. Svi sustavi za upravljanje bazama podataka nastoje što više slijediti originalni standard, ali također pokušavaju obogatiti SQL različitim opcijama koje nisu standardizirane. Zato se često mogu uočiti razlike u sintaksi SQL naredbi koje se koriste u različitim bazama podataka[4][5].

Podjela SQL-a i osnovne naredbe

- SQL se dijeli na četiri "podjezika":
- Data Definition Language (DDL) sastoji se od naredbi za definiranje objekata baze podataka: npr. CREATE, DROP i ALTER
- Data Manipulation Language (DML) sastoji se od naredbi za manipuliranje podacima u bazi: npr. INSERT, DELETE, UPDATE
- Data Control Language (DCL) sastoji se od naredbi za kontrolu podataka i njihovu sigurnost: npr. GRANT i REVOKE
- Data Query Language (DQL) sastoji se od naredbi za pretraživanje i dohvaćanje podataka u bazi: npr. SELECT

Naredba SELECT jedna je od najkorištenijih SQL naredbi, a njena snaga naročito dolazi do izražaja upotrebom različitih operatora, funkcija i ključne riječi WHERE. Tome će se kasnije posvetiti čitavo poglavlje, ali prije toga će se u idućem dijelu govoriti o tipovima podataka u SQL-u [5].

3.2. Baza podataka za web aplikaciju

Baza podataka bit će jedan od bitnijih djelova kako bi ova web stranica postala aplikacija, odnosno napraviti će se mjesto gdje će se spremati svi podaci koji će biti uneseni. Sam tijekom izrade baze podataka nije težak postupak. Baza podataka izrađuje se uz pomoć programa „phpMyAdmin“ koji se pokreće u web pregledniku te dodatnog programa „XAMPP“ koji će izvesti simulaciju servera na računalu. U nastavku će se detaljno prikazati sama izrada baze podataka.

3.2.1. Izrada baze podataka

Početak izrade obavlja se upisivanjem „localhost/phpmyadmin“-a u web preglednik kako bi se pristupilo programu. Unutar programa vidljivi su alati za izradu baze podataka stoga slika 3.1. prikazuje početak izrade te unosenje imena baze podataka.



Sl. 3.1. Početak izrade baze podataka

Nastavak izrade baze podataka svodi se na unošenje nove tablice te broj željenih kolona, taj postupak prikazan je na slici 3.2. Prema slici 3.3. vidljive su sve kolone te po želji odabrane metode unosa što čini najvažniji dio baze podataka. Kako bi sve informacije bile na svom mjestu te da ne dođe do neželjenih pogrešaka metode je potrebno definirati.

U bazi podataka nisu pronađene tablice

Izradi tablicu

Naziv: evidencija Number of columns: 7

Kreni

Sl. 3.2. Unos tablice

Naziv	Vrsta	Duljina/Vrijednosti	Zadano	Uspoređivanje	Atributi	Null	Indeks	A_I	Komentari	MIME vrsta
lokacija	VARCHAR	20	None			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>		
dan	DATE		None			<input type="checkbox"/>	---	<input type="checkbox"/>		
radnici	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>		
grupa	INT	10	None			<input type="checkbox"/>	---	<input type="checkbox"/>		
vrijeme_dolaska	TIME		None			<input type="checkbox"/>	---	<input type="checkbox"/>		
vrijeme_odlaska	TIME		None			<input type="checkbox"/>	---	<input type="checkbox"/>		
opis_rada	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>		

Komentari tablice:

Pogon pohrane: InnoDB Uspoređivanje:

Definicija PARTICIJE:

Spremi

Sl. 3.3. Unošenje vrijednosti pojedinih elemenata

Daljnji nastavak izgleda baze podataka prikazuje cjelokupnu strukturu baze podataka koja je prikazana direktno u programu. Slika 3.4. detaljno prikazuje strukturu baze podataka.

Pretraživanje									
Strukturu									
SQL									
Traži									
Umetni									
Izvoz									
Uvoz									
Operacije									
Praćenje									
Okidači									
#	Naziv	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno	Aktivnost	
<input type="checkbox"/>	1 lokacija	varchar(20)	latin1_swedish_ci	Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	2 dan	date		Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	3 radnici	varchar(100)	latin1_swedish_ci	Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	4 grupa	int(10)		Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	5 vrijeme_dolaska	time		Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	6 vrijeme_odlaska	time		Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	
<input type="checkbox"/>	7 opis_rada	varchar(100)	latin1_swedish_ci	Ne	None			Promijeni Ispusti Primarni Jedinstveno Indeks Spatial Puni tekst Distinct values	

☐ Označi sve S odabirom: ☐ Pretraživanje Promijeni Ispusti Primarni Jedinstveno Indeks

Sl. 3.4. prikazuje strukturu podataka

Baza podataka, odnosno sam program nudi direktno unošenje podataka u kolone što detaljno prikazuje slika 3.5., no unutar ovoga projekta razvija se sustav koji bi putem web aplikacije, a ne putem same baze podataka, unio podatke u bazu podataka. Dosada je prikazana samo jedna tablica odnosno tablica za evidenciju, no baza se sastoji od još jedne tablice koja bi trebala poslužiti kao „login“ odnosno evidencija korisničkih imena i lozinki kako bi administracija mogla pristupiti aplikaciji.

Stupac	Vrsta	Funkcija	Null	Vrijednost
lokacija	varchar(20)	<input type="text"/>		<input type="text"/>
dan	date	<input type="text"/>		<input type="text"/>
radnici	varchar(100)	<input type="text"/>		<input type="text"/>
grupa	int(10)	<input type="text"/>		<input type="text"/>
vrijeme_dolaska	time	<input type="text"/>		<input type="text"/>
vrijeme_odlaska	time	<input type="text"/>		<input type="text"/>
opis_rada	varchar(100)	<input type="text"/>		<input type="text"/>

Kreni

Sl. 3.5. polje za unos podataka

Polja prikazana na slici 3.5. bit će popunjena informacijama putem web aplikacije, a ne direktnim unosom, dok će samo brisanje podataka biti moguće unutar baze podataka, ostali unosi su putem web aplikacije. Slika 3.6. prikazuje polja pri izradi druge tablice "Login" koja služi za unošenje podataka korisničkog imena i lozinke, a ti podaci bit će izravno uneseni u bazu podataka što je prikazano na slici 3.7. Izravan unos podataka za spajanje mora se izvršiti kako bi

se administrator mogao spojiti na web aplikaciju jer je sama web aplikacija spojena s bazom i ne dopušta ulazak u web aplikaciju ako se korisničko ime i lozinka ne podudaraju.

Naziv	Vrsta	Duljina/Vrijednosti	Zadano	Uspoređivanje	Atributi	Null	Indeks	A_I	Komentari	MIME vrsta
korisnik	INT	20	None				PRIMARY			
lozinka	INT	20	None							

Komentari tablice:

Pogon pohrane: Uspoređivanje:

Definicija PARTICIJE:

Sl. 3.6. polja tablice "login"

Slika 3.7. prikazuje unos podataka za login, odnosno za povezivanje i pristup aplikacije s bazom podataka kako bi se odobrio pristup unošenju podataka u polja web aplikacije. Web aplikacija radi sigurnosnih razloga ne dopušta spajanje podacima koji nisu u bazi podataka, odnosno u tablici "login". Korisničko ime je pod "int" što znači da je korisničko ime sastavljeno od brojki. Svakom administratoru se dodjeljuju brojke što nije klasičan pristup administratora, već novi način raspodjele po brojevima. Slika 3.8. prikazuje spremljene podatke unutar tablice "login" i pristupanjem web aplikacije prihvaća se samo korisničko ime i lozinka iz te tablice jer je putem PHP - programskog jezika povezana izravno u HTML kodu, što je obliže objašnjeno u sljedećem poglavlju.

Stupac	Vrsta	Funkcija	Null	Vrijednost
korisnik	int(20)	<input type="text"/>		0013
lozinka	int(20)	<input type="text"/>		1234

☒ Ignoriraj

Stupac	Vrsta	Funkcija	Null	Vrijednost
korisnik	int(20)	<input type="text"/>		
lozinka	int(20)	<input type="text"/>		

Sl. 3.7. Prikaz unosa podataka u tablicu "login"

```
SELECT *
FROM `login`
LIMIT 0, 30
```

Prikaži : Start row: Number of rows: Headers every rows

+ Opcije

	korisnik	lozinka
<input type="checkbox"/> Uređivanje <input type="checkbox"/> Kopiraj <input type="checkbox"/> Izbriši	13	1234

☐ Označi sve
 S odabirom:
 ☐ Promijeni
 ☐ Izbriši
 ☐ Izvoz

Prikaži : Start row: Number of rows: Headers every rows

Sl. 3.8. Podaci administratora

3.3. Prikaz strukture baze podataka

Tab. 3.1. Prikaz strukture tablice „evidencija“

Field	Type	Null	Default
lokacija	varchar(20)	No	
dan	date	No	
radnici	varchar(100)	No	
grupa	int(10)	No	
vrijeme_dolaska	time	No	
vrijeme_odlaska	time	No	
opis_rada	varchar(100)	No	

Tablica 3.1. prikazuje strukturu baze podataka odnosno strukturu tablice evidencija. Sama tablica se bazira na traženim informacijama, odnosno na spoju s web aplikacijom kako bi se podaci mogli spremiti u bazu nakon stiska na gumb "upis", što će detaljno biti pojašnjeno u

nastavku rada. Tablica 3.2. prikazuje strukturu podataka tablice "login" koja u sebi sadrži stavke korisničkog imena i lozinke.

Tab.3.2. prikaz strukture tablice „login“

Field	Type	Null Default
korisnik	int(20)	No
lozinka	int(20)	No

3.4 SQL naredbe za izradu i popunjavanje baze podataka

Niže navedene SQL naredbe predstavljaju, naredbe za kreiranje baze podataka, naredbe za kreiranje tablice „login“ te naredbe za kreiranje tablice „evidencija“.

Kreiranje baze podataka:

```
CREATE DATABASE Vinograd;
```

Kreiranje tablice „login“:

```
create table login (  
    korisnik int(20) not null,  
    korisnik int(20) not null  
);
```

Kreiranje tablice „evidencija“:

```
create table evidencija (  
    lokacija varchar(100) not null primary key,  
    dan date not null,  
    radnici varchar(20) not null,
```

```
grupa int (10) not null,  
vrijeme_dolaska time not null,  
vrijeme_odlaska time not null,  
opis_rada varchar(100) not null,  
);
```

4. PHP skriptni jezik

PHP je akronim za **PHP: Hypertext Preprocessor**. PHP je skriptni programski jezik. Iako je svoj razvoj započeo kao „hypertext preprocessor“, danas je on uvelike proširio svoje mogućnosti i ta definicija se gotovo više ni ne koristi. Osim kao podrška web aplikacijama danas je PHP moguće koristiti i kao konzolnu aplikaciju, ali je jednako moguće pisati i aplikacije u PHP-u s punim grafičkim korisničkim sučeljem, upotrebljavati OpenGL biblioteke za trodimenzionalnu vizualizaciju i slično. Danas je PHP vrlo moderan programski jezik koji je najčešće u upotrebi upravo na webu.

PHP počinje u rukama programera Rasmusa Lerdorfa koji je i danas glavna osoba zadužena za njegov razvoj. PHP je u početku tek bila pomoć Rasmusu Lerdorfu za izradu vlastite web stranice tako što je s nekoliko CGI programa pisanih u programskom jeziku C zamijenio programe pisane u programskom jeziku Perl koje je do tada koristio. Inačica 1 i 2 nisu bile previše popularne, ali inačica 3, čija je konačna inačica izašla 1998. godine, privukla je velik broj poklonika ovog novog programskog jezika. 2000. godine izlazi i inačica 4 da bi 2004. izašla aktualna inačica 5 koja uključuje brojne novosti kao što su uključena podrška za Unicode znakove, objektno orijentirani pristup, podrška za XML, poboljšana podrška za pristupanje MySQL bazi podataka, uključenu podršku za upotrebu Web servisa kao i noviju inačicu optimizatora koda Zend čime su postignute znatno bolje performanse samog koda[6].

Velik broj osnovnih funkcija ima istu sintaksu. Početnicima je također vrlo dobar jer nije ograničavajući kao neki drugi programski jezici budući da nema deklaracija tipova varijabli, a moguće je i istu varijablu koristiti za pohranu različitih vrsta vrijednosti. Iako ove karakteristike programski kod čini manje osjetljivim na pogreške, jednostavnije ih je i napraviti budući da prevoditelj (eng. Compiler) na njih neće upozoriti. Danas se PHP koristi na nešto manje od

dvadeset milijuna različitih web stranica prema istraživanju tvrtke Netcraft. Programe pisane u programskom jeziku PHP nije potrebno prevoditi u izvršni oblik (eng. Compile), već se oni izvode prilikom pokretanja u interpreteru. Programi se pišu kao dio HTML stranice. Sam program nalazi se unutar HTML oznake koji počinje s `<?PHP`, a završava s `?>`. Sve unutar ove oznake se smatra PHP programom i ako je na poslužitelju instaliran i ispravno podešen PHP interpreter, a datoteka završava s `.php` tada će se taj dio programa automatski izvršiti [5][6].

4.1. Projekt

Kako bi sama web aplikacija dobila svoj značaj potrebno je integrirati PHP kod koji će povezati samu stranicu s bazom te dodijeliti im funkcionalnosti. Ovaj projekt zasniva se na temelju programskog jezika PHP koji će u krajnosti web stranicu pretvoriti u web aplikaciju. Nastavak ovoga dokumenta detaljno će prikazati rad web aplikacije nakon formiranja i integriranja PHP formi unutar samoga HTML okruženja.

4.2. Osposobljavanje web aplikacije

Za osposobljavanje odnosno integriranje funkcionalnog dijela u web stranicu koristi se PHP programski jezik koji je ranije opisan u prvom dijelu istog poglavlja. Sam PHP kod prepoznatljiv je u HTML kodu po svom označavanju koji je sljedeći:

`"<?php"` - za početak korištenja te `"?>"` - za kraj korištenja.

PHP funkcionira na pozivanju odnosno pokretanju pojedinih sesija ("session") koje dovode do povezivanja odnosno određivanja putanji do baze podataka. Ovaj projekt baziran je na korištenju "localhost"-a, a time i rad programa XAMPP bez kojega sam projekt ne bi bio moguć niti izvediv jer XAMPP koristi osobno računalo kako bi simulirao rad poslužitelja. Cjelokupan kod rađen je po raznim tutorialima i pojašnjenjima funkcionalnosti PHP jezika te uz pomoć programa "NetBeans" koji se pokazao kao vrlo učinkovit program za programiranje, no tijekom učenja i istraživanja započeto je korištenje programa "Notepad++" koji se također pokazao vrlo učinkovitim te mnogo jednostavnijim za korištenje.

Nastavak dokumenta prikazuje pojedine djelove koda pri integriranju u stilski jezik te je slikovito popraćena svaka pojedina radnja koja ukazuje na funkcionalnost same web aplikacije, a ujedno je ovo jezgra cijeloga projekta koji daje krajnje rezultate koji će biti vidljivi na slikama.

5 . Web aplikacija

Sam projekt započinje sa stilskim izgledom same aplikacije, koja se kreira uz pomoć navedenih tehnologija HTML i CSS, koje će svojim doprinosom prikazati jednu sofisticiranu web aplikaciju, koja se može koristiti kao evidentiranje sezonskih radova. Sam izgled se bazira na HTML-u te pozivanjem CSS stila unutar HTML-a dolazi do konačnog izgleda web aplikacije. U nastavku će se prikazati sam postupak tijekom izrade web aplikacije, odnosno detalje iz koda koje će se pomno popratiti i pojasniti kako bi se objasnio postupak izrade.

```
<h3>Prijava se</h3>
<div id="stranica">
  <div id="stranica-bggore">
    <div id="stranica-bgg">
      <div id="sadrzaj">
        <div class="student">
          <h2 class="naslov">web aplikaciju koja omogućava evidentiranje sezonskih</h2>
          <form action="login.php" method="post">
            <p><td>Korisnicko ime:</td>
              <td><input type="text" name="username" maxlength="20" /></td>
            </tr>
            <p><tr>
              <td>Lozinka:</td>
              <td><input type="password" name="password" maxlength="10" /></td>
            </tr></p>
            <p><tr>
              <td colspan="2"><input type="submit" name="submit" value="Prijava" /></td>
            </tr>
          </form>
```

Sl. 5.1. Prikazuje uređivanje html koda

Slika 5.1. prikazuje izgled web stranice odnosno aplikacije (kodni oblik) gdje su prikazani svi detalji, koji su ranije navedeni u uvodu, dakle od samih formi do postavljanja naslova (h1...h6) pa do otvaranja pojedinih „div“ elemenata koji onda raspoređuju redom elemente po pojedinim blokovima.

Slika 5.2. prikazuje pozivanje CSS stilske jezika unutar HTML-a. Unutar „head-a“ se poziva „style.css“ koji dodjeljuje konačan izgled blokova odnosno koji dodaje boju te veličinu i centriranje (da se pojedini element nalazi lijevo, desno ili u sredini).

```

<head>

    <title>Početna</title>

    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />

    <meta name="keywords" content="Site keywords here" />
    <meta name="description" content="Site description here" />
    <meta name="author" content="Your name here" />

    <link rel="stylesheet" type="text/css" href="css/style.css" />

</head>

```

Slika 5.2. Prikaz pozivanja CSS-a unutar HTML-a

Nakon što je prikazan način pristupanja CSS-u odnosno datoteki „style.css“ prikazuje se sam izgled koda unutar CSS datoteke. Primjetno je da se sam kod svodi na boju koja se prikazuje u formatu od 6 znakova. Pri izradi samih stilova postoje generatori koji daju točan poredak simbola za svaku boju. Slika 5.3. prikazuje sam izgled koda. Slika prikazuje detaljan dio koda koji pojašnjava ranije navedene pozicije te veličine.

```

#head {
    width: 700px;
    height: 150px;
}

#head h1 {
    margin: 0;
    padding: 0;
    display: block;
    width: 700px;
    height: 150px;
    background-image: url(../images/head.jpg);
    text-indent: -10000px;
}

/* Navigation */

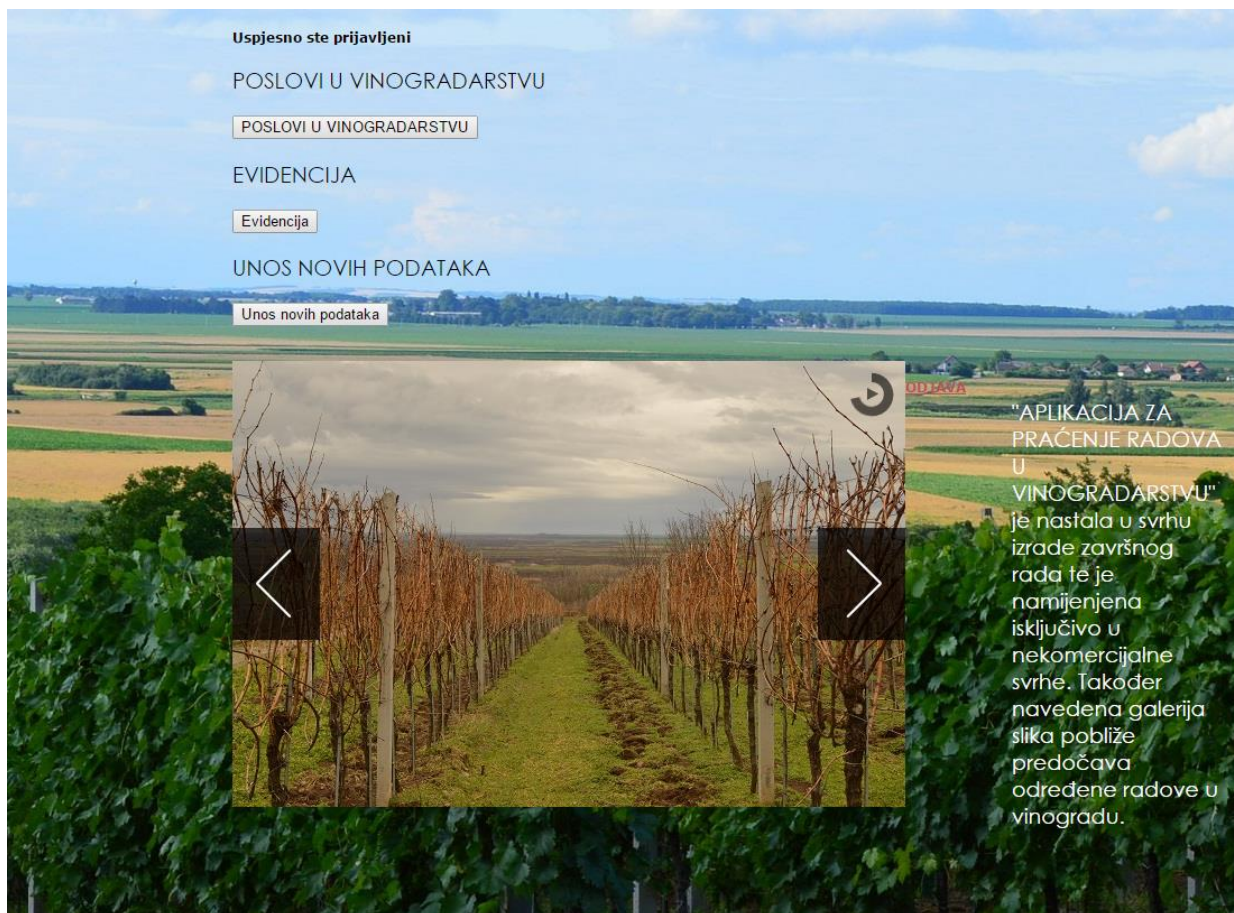
#nav {
    margin: 0;
    padding: 0;
    width: 700px;
    height: 20px;
}

```

Sl. 5.3. Dio CSS koda

Nakon prikaza izgleda web stranice u kodnom obliku te samog uređivanja i pisanja koda, dobiva se jedna web stranica lijepog izgleda koja ne samo da ima svoju namjenu, već i dobar

izgled, što je važno u svakom poslu web dizajnera - napraviti web stranicu što boljeg i modernijeg izgleda. Slika 5.4. prikazuje krajnji izgled spoja dvaju stilskih jezika, odnosno prikazuje krajnje rješenje (stilsko) koje će zadovoljiti korisnike aplikacije.



Sl. 5.4. Konačni izgled web stranice/aplikacije

Nakon cjelokupnog opisa stilskog izgleda vidljive su pojedine rubrike za upis pojedinih traženih informacija, stoga nastavak ovoga dokumenta opisuje bazu podataka koja će sadržavati sve podatke te PHP programski jezik koji je integriran u HTML-kodu kako bi sama web aplikacija bila funkcionalna i povezana s bazom podataka.

5 . 1. Login i unos podataka u bazu

Početak je kod same naslovne stranice novoizrađene web stranice na osnovu teme projekta gdje se zahtjeva povezivanje odnosno tzv. "login" administratora, čime je potrebno dodijeliti formu za login, odnosno integrirati putanju "login.php" u HTML dokumentu. Slika 5.5.

detaljno prikazuje sam izgled forme unutar stilskog jezika te je u nastavku (ispod slike) prikazana forma logina koja je rađena po tutorialima kojih je mnogo.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<?php
{
require("login.php");
}
?>
<head>
```

SI.5.5. Obrazac unutar *login.php* skripte

Prikazan je dio forme "login.php" unutar kojega se vidi logika rada PHP jezika bazirana na pozivanju sesija te pozivanju drugih .php dokumenata putem kojih se ostvaruje samo spajanje na bazu podataka.

```
<?php require_once("session.php"); ?>
<?php require_once("connection.php"); ?>
<?php require_once("functions.php"); ?>
<?php
include_once("form_functions.php");
session_start();
if (isset($_POST['submit'])) {
    $errors = array()
    $required_fields = array('korisnik', 'lozinka',);
    $errors = array_merge($errors, check_required_fields($required_fields, $_POST));

    $fields_with_lengths = array('korisnik' => 20, 'lozinka' => 20, );
    $errors = array_merge($errors, check_max_field_lengths($fields_with_lengths, $_POST));

    $korisnik = trim(mysql_prep($_POST['korisnik']));
    $lozinka = trim(mysql_prep($_POST['lozinka']));
```

Vidljivo je da se traže lozinka i korisnik kako bi došlo do pravovaljanog spajanja, podaci se moraju podudarati s podacima u bazi podataka koji su pozvani funkcijom "connecting". Ta funkcija unutar sebe sadrži tzv. "constants" odnosno poziva konstantu unutar sebe.

Izgled "connstants.php" funkcije:

```
<?php
define("DB_SERVER", "localhost");
define("DB_USER", "root");
define("DB_PASS", "");
define("DB_NAME", "Vinograd");
?>
```

Kao što je prikazano u dijelu koda poziva se spajanje na bazu podataka "Vinograd" koji unutar sebe sadrži sve informacije bitne za povezivanje na web aplikaciju te putem koje će se spremati podaci u navedenu bazu podataka.

Na slici 5.6. prikazat će se integrirani PHP kod unutar stilskog jezika. Detaljno se opisuje struktura za unošenje podataka odnosno spremanje istih kako bi funkcionalnost bilo na najvećoj razini. Prikazani su svi traženi segmenti, PHP kod spojen je s bazom podataka tako da bi se spremanje trebalo automatski izvršiti, što bi trebalo biti prikazano u bazi podataka, kao rezultat cjelokupnog projekta.

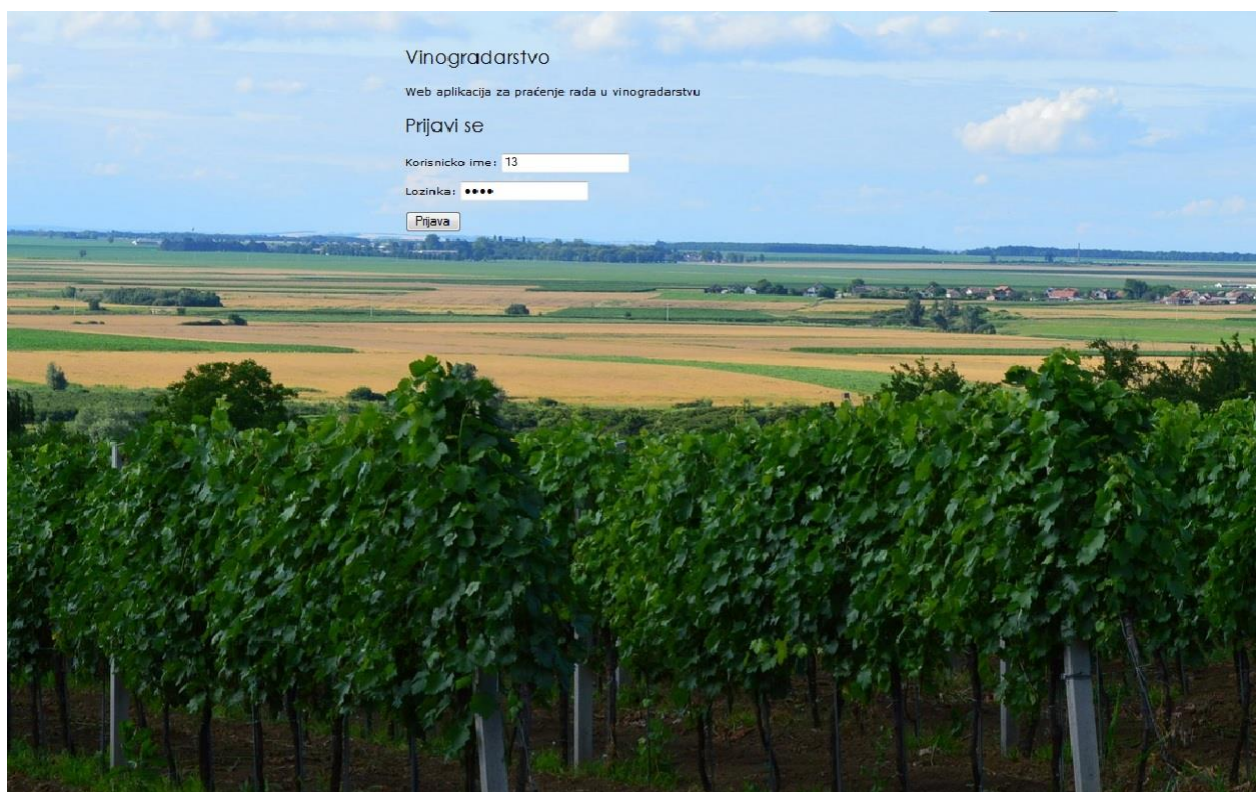
```
?>
<!-- forma za unos -->
<form method="post" action="">
<p>lokacija: <input type="text" name="lokacija"></p>
<p>dan: <input type="text" name="dan"></p>
<hr>
<p>radnici: <input type="text" name="radnici"></p>
<p>grupa:
<select name="grupa">
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>
</select></p>
<form method="post" action="">
<p>vrijeme dolaska: <input type="text" name="dol"></p>
<p>vrijeme odlaska: <input type="text" name="odl"></p>

<hr>
<p>opis rada :<br><textarea name="opisrada"></textarea><br>
<br><br>
<input type="submit" name="spremanje" value="upis">
<hr>
</form>

<?php
```

Sl. 5.6. PHP unutar stilskog jezika

Kodni oblik na slici 5.6. prikazan je u stvarnom obliku koji je svim korisnicima vidljiv, a na slici 5.8. sve što je definirano stilskim izgledom i povezano pomoću PHP jezika u vizualnom obliku pokrenutom od strane web preglednika. Kako bi došlo do mogućnosti unosa podataka najprije je potrebno obaviti povezivanje odnosno tzv. "login" na web aplikaciju. Unošenje korisničkih podataka odnosno login prikazan je na slici 5.7.

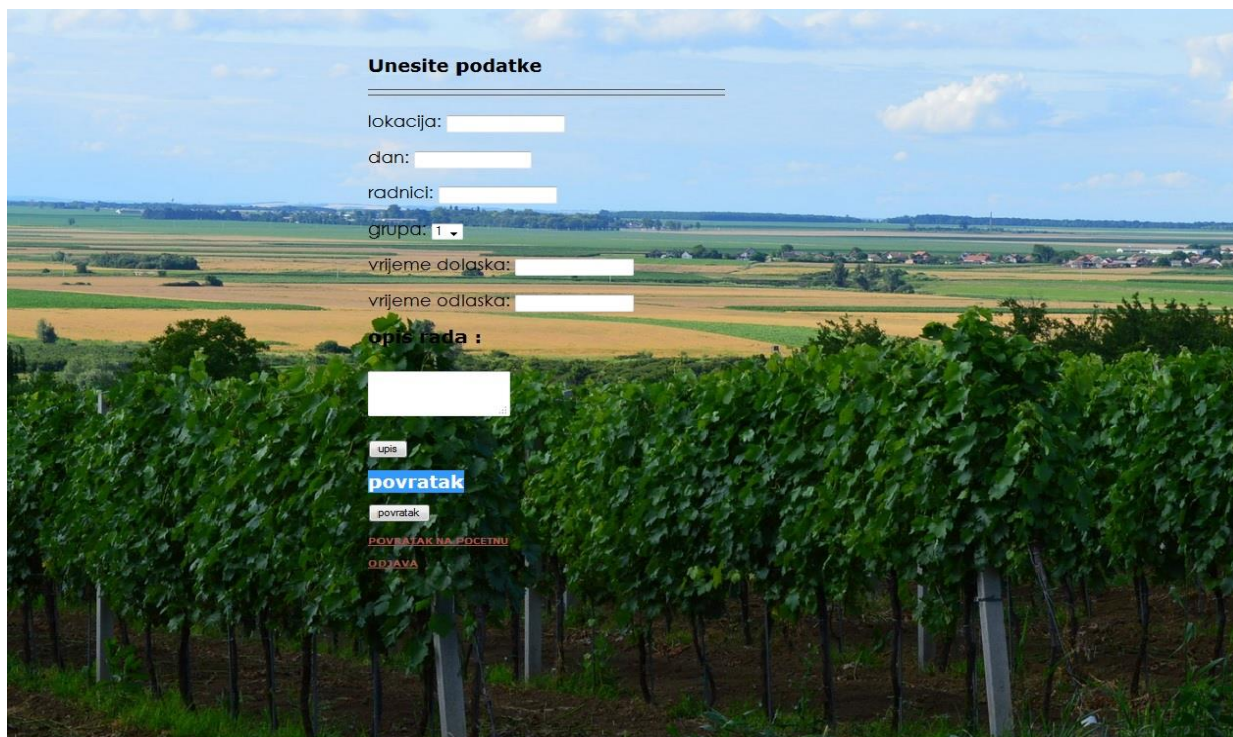


Sl. 5.7. Unošenje korisničkih/administratorskih podataka

Korisnički su podaci, kao što je ranije navedeno, brojčani. Unutar baze podataka dodijeljeni su korisničko ime i lozinka. Za pravovaljan rad s aplikacijom potrebna je točnost pri povezivanju odnosno unošenje postojećih, pravovaljanih podataka. Ako dođe do pogreške odnosno ako se ne unesu točni podaci tijekom spajanja na web aplikaciju, web stranica će evidentirati krivo unošenje podataka te korisnik neće imati dopuštenje rada s aplikacijom.

Nakon unošenja korisničkih informacija nastavak projekta zasniva se na traženim stavkama, odnosno nakon povezivanja s aplikacijom ponuđene su stranice za novo unošenje podataka što je prikazano na slici 5.9. te mogućnost odabira evidencije što je prikazano na slici

5.10. Stranica koja nudi evidenciju uređena je od strane voditelja web aplikacije što u konačnici samo daje opis radova i prijava poteškoća, ako ih je bilo.



Unesite podatke

lokacija:

dan:

radnici:

grupa: 1 ▾

vrijeme dolaska:

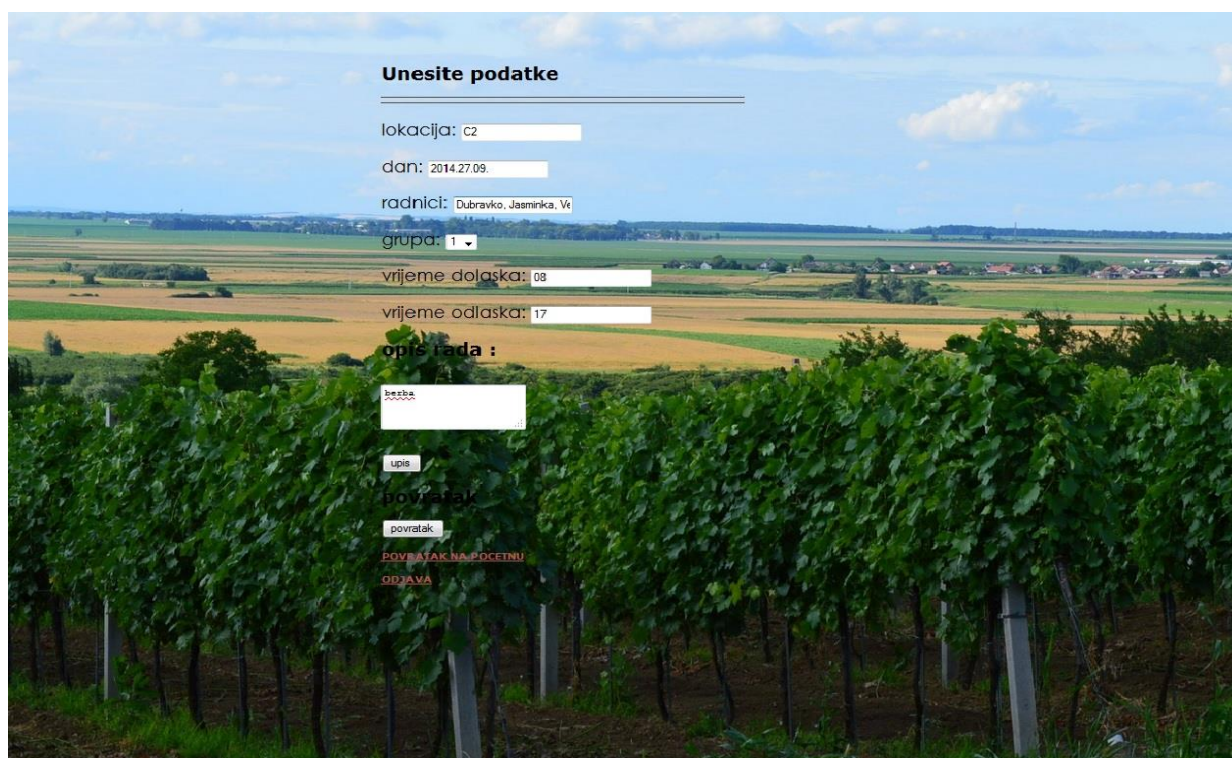
vrijeme odlaska:

opis rada :

[POVRATAK NA POČETNU](#)

[ODJAVA](#)

Sl. 5.8. Polja za unos podataka



Unesite podatke

lokacija: c2

dan: 2014.27.09.

radnici: Dubravko, Jasminka, Vi

grupa: 1 ▾

vrijeme dolaska: 08

vrijeme odlaska: 17

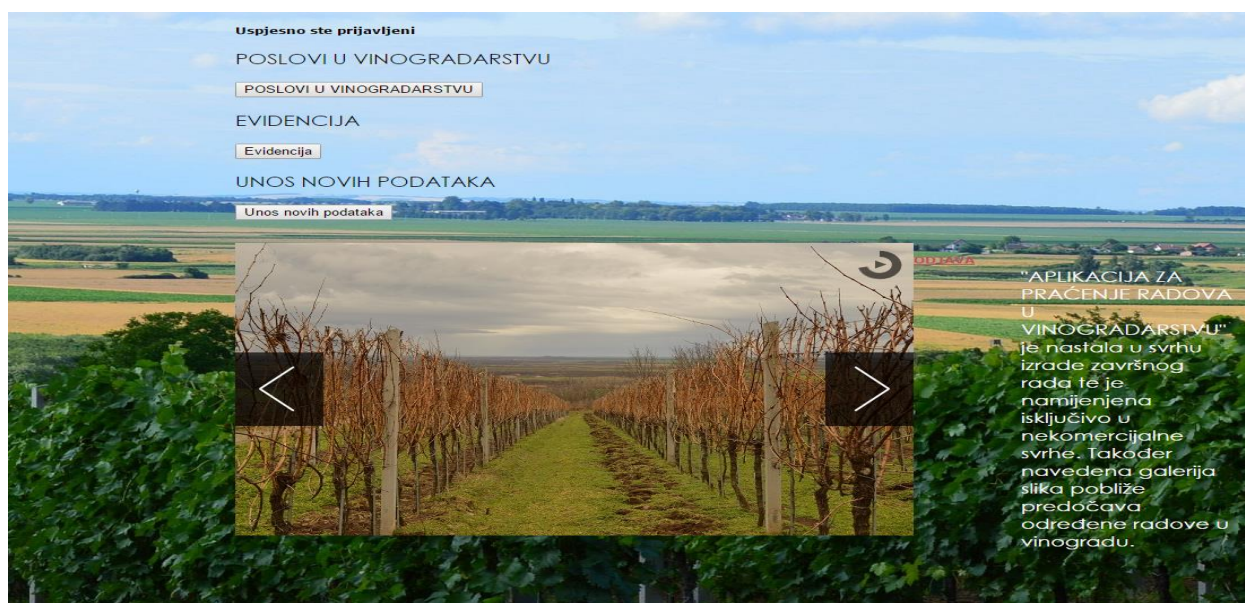
opis rada :

[POVRATAK NA POČETNU](#)

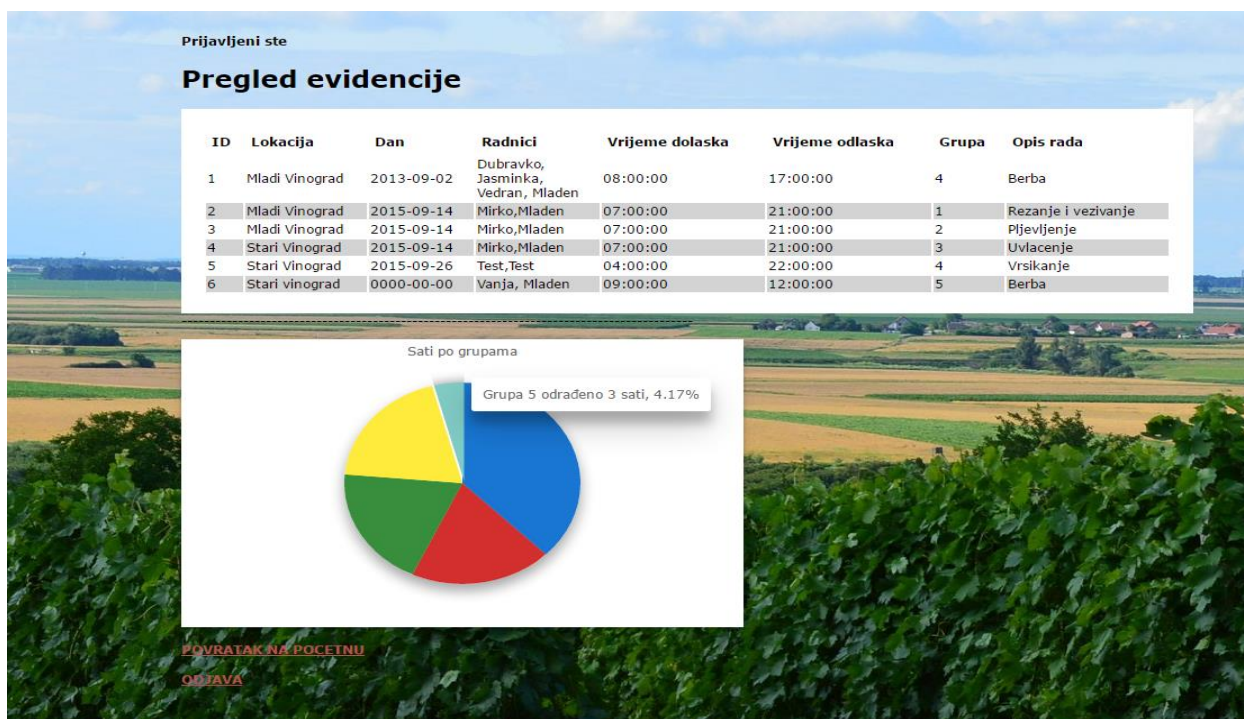
[ODJAVA](#)

Sl. 5.9. Unos podataka

Pri odabiru željene radnje, nakon login stranice, ponuđen je prijelaz na stranice različitog tipa. Prva je poslovi u vinogradarstvu s kratkim opisom svakog posla dok druga stranica nudi samo prikaz informacija odnosno kratak sažetak o stanju i praćenju rada, a iduća nudi upis podataka što je već ranije navedeno i prikazano u tekstu. Stranica koja nudi prikaz mogućnosti prikazana je na slici 5.10. te stranica koja prikazuje informacije o praćenju tijekom rada prikazana je na slici 5.11.



Sl. 5.10. Prikaz stranice odabira



Sl. 5.11. Stranica praćenja rada

Sve stranice usklađene su po potrebama korisnika, a da bi sve bilo funkcionalno potrebno je provjeriti bazu podataka u kojoj bi trebale biti upisane informacije nakon klika na gumb upisivanja. Slika 5.12. prikazuje evidentirane informacije unutar baze podataka te je time sam rad aplikacije neupitan. Stranica koja ispisuje evidenciju nije povezana s bazom podataka te je uređena od strane administratora web stranice.

+ Opcije

</

Sl. 5.12. Upisani podatci u bazi podataka

6. ZAKLJUČAK

Ovaj projekt nastao je kao rezultat zahtjeva i učenja te izrade završnog rada. Rad se temelji na osnovnim tehnologijama koje se koriste pri izradi web stranice te osnovne tehnologije izrade baze podataka. Navedene dvije tehnologije u konačnici se povezuju kako izrađena web stranica ne bi bila samo stilski uređena i informativna web stranica, već i web aplikacija s pojedinim mogućnostima. Web aplikacija ovakvoga tipa služi kao pokazatelj mogućnosti izrade različitih namjena. Namjena ovoga projekta je, u prvom redu, da bude pokazatelj naučenih tehnologija tijekom studiranja dok se sama svrha temelji na temi samoga rada kako bi se doprinijelo u svim segmentima društva i rada pa tako i u vinogradarstvu.

Korištene tehnologije baziraju se na otvorenom kodu, odnosno besplatne su za korištenje te time ne ugrožava ničija prava. Tehnologije su se koristile i tijekom studiranja u manjoj mjeri, ali su izradom projekta ovakve vrste dobile novo značenje i time još veću želju i mogućnost u daljnjem korištenju istih.

LITERATURA

- [1] <http://hr.wikipedia.org/wiki/HTML> 20.05.2016. 17:00
- [2] <http://www.saluto.hr/index.php/html-programski-jezik-osnova-razvoja-web-stranica.html> 20.05.2016. 17:00
- [3] <http://www.saluto.hr/index.php/css-stilski-jezik-oblikovanje-web-stranica.html>
- [4] https://hr.wikipedia.org/wiki/Baza_podataka 23.05.2016. 13:00
- [5] <http://www.webmajstori.net/clanci/programiranje/uvod-u-baze-podataka-i-sql/92/> 23.05.2016. 13:00
- [6] <http://www.scribd.com/doc/51344155/9/Programski-jezik-PHP> 25.05.2016. 08:00
- [7] <http://php.com.hr/68> 25.05.2016. 08:00
- [8] <https://php.net/manual/en/> 28.05.2016. 15:00
- [9] <http://www.textfixer.com/tutorials/html-search-box.php> 28.05.2016. 15:00
- [10] <http://www.script-tutorials.com/advance-php-login-system-tutorial/> 01.06.2016. 17:00
- [11] <http://www.mysql.com/> 01.06.2016. 17:00
- [12] <http://www.w3schools.com/css/> 02.06.2016. 10:00
- [13] <http://www.w3schools.com/php/> 03.06.2016. 10:00
- [14] Osnove Internet Programiranja/ Ivica Lukić, Mirko Köhler. – Osijek: Elektrotehnički fakultet, 2011. 20.05.2016. 10:00

SAŽETAK

Ovaj projekt prikazuje izradu web stranice/aplikacije koja omogućava evidentiranje sezonskih poslova u vinogradarstvu. Za samu izradu web aplikacije korišteni su programi "phpMyAdmin", "Notepad++", "Google Chrome" te program "XAMPP". Tehnologije koje su uspješno korištene, u ovom radu, za stilski izgled web stranice su "HTML" i "CSS" te programski jezik "PHP" pomoću kojega je uspješno izvršeno spajanje na bazu podataka koja je napravljena korištenjem tehnologije "SQL" te time postala aplikacija.

Ključne riječi: CSS, HTML, PHP, SQL.

ABSTRACT

Web applications for managing vineyard economy

This project shows the development of web site / application that enables recording of seasonal jobs in viticulture. For the purpose of making web applications used programs "phpMyAdmin", "Notepad + +", "Google Chrome" and the program "XAMPP". Technologies that have been used successfully, in this paper, the stylish look of the website are "HTML" and "CSS" and the programming language "PHP" by means of which he successfully completed connecting to the database that was created using the technology, "SQL", and thus become applications.

Key words: CSS, HTML, PHP, SQL.

ŽIVOTOPIS

Mladen Krušarovski rođen je 13.04.1990. godine u Osijeku. Osnovnu školu 2005. godine završio je u OŠ Dr. Franje Tuđmana te nakon toga upisao tehničku školu u Belom Manastiru, smjer tehničar za računalstvo. Po završetku srednje škole, 2009. godine, upisao je Elektrotehnički fakultet u Osijeku, stručni studij elektrotehnike – smjer informatika.

Potpis:
