Web aplikacija-Menadžersko praćenje poslovanja tvrtke

Balaš, Igor

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:200:770975

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij

WEB APLIKACIJA – MENADŽERSKO PRAĆENJE POSLOVANJA TVRTKE

Završni rad

Igor Balaš

Osijek, 2016. godine

SADRŽAJ

1. Uvod 1

1.1. Zadatak završnog rada 1

2. Tehnologije 2

2.1. HTML 2

2.2. Javascript 2

2.3. PHP 3

2.4. CSS 3

2.5. JQUERY 3

2.6. XAMPP 4

2.7. Microsoft SQL 4

3. Razvoj aplikacije 6

3.1. Knjigovodstvene evidencije i praćenja 6

3.2. Baza podataka 6

3.3. Sučelje 9

4. Zaključak 22

Literatura 23

Sažetak 24

Abstract 25

Životopis 26

1. UVOD

Na temelju dugogodišnjeg iskustva u projektiranju i izradi knjigovodstvenih sustava, a razvojem komunikacijskih tehnologija, uočena je potreba izrade web aplikacije koja je namijenjena direktorima i menadžerima u tvrtkama koji često izbivaju izvan prostora tvrtke, pa samim time i izvan lokalnog informatičkog sustava. Aplikacija pruža mogućnost uvida u knjigovodstvene podatke u realnom vremenu i na taj način doprinosi u znatno kvalitetnijem donošenju poslovnih odluka. Korisnici aplikacije su često u pokretu, pa izrada ovakve vrste aplikacije omogućava pristup podacima na mobilnim uređajima poput pametnih telefona i tableta, uz pretpostavku raspoloživosti pristupa komunikacijskim uslugama za prijenos podatka, odnosno Internetu. Izrada aplikacije temelji se na već postojećoj bazi podataka koja se nalazi u lokanom sustavu i koja se redovno ažurira dnevnim podacima od strane djelatnika u knjigovodstvenoj službi. U takvom sustavu potrebno je osigurati pristup bazi podataka sa udaljenog mjesta putem Interneta vodeći računa da sigurnosne pretpostavke i dalje budu zadovoljene. Već postojeća rješenja su izrađena komercijalnim alatima kao što je u slučaju baze podataka Microsoft SQL Server. Obzirom da postoji niz alata koji su otvorenog tipa (engl. *open source*) isti su korišteni za potrebe izrade ove aplikacije. Potreba za brzim pristupom određenim podacima kao što su kartice kupaca i dobavljača, financijske kartice, otvorene stavke, bilanca, pregled prihoda i rashoda, pretpostavlja izradu aplikacije sa manjim brojem najčešće korištenih pregleda kako bi bila što jednostavnija pri uporabi. Na toj osnovi, u komercijalizaciji aplikacije, moguće je vrlo jednostavno napraviti željena proširenja i personalizirati istu u skladu sa specifičnosti određenih djelatnosti pojedinih tvrtki.

U slijedećim poglavljima detaljnije će biti opisane korištene tehnologije, struktura baze podataka, programski jezici i svi potrebni alati.

1.1. Zadatak završnog rada

Nastavno na uvod može se zaključiti da je zadatak završnog rada izrada web aplikacije za menadžersko praćenje poslovanja tvrtke korištenjem web tehnologija i postojeće baze podataka. Osigurati uvid u podatke u realnom vremenu na temelju kojih se mogu donositi kvalitetne poslovne odluke. Baza podataka je Microsoft SQL Server, a potrebno je koristiti slijedeće web tehnologije: HTML5, CSS3, PHP, Javascript, JQuery i dr.

2. TEHNOLOGIJE

2.1. HTML

Hyper Text Markup Language je jezik za opis web stranica. HTML nije programski jezik i koristi oznake (engl. *tags*) za opis sadržaja. Oznake uglavnom dolaze u paru (npr. <*b*><*/b*>) i jedna označava početak a druga kraj. Između njih se obično dodaje tekst i atributi sa željenim vrijednostima. Internetski preglednici (engl. *browser*) interpretiraju zapisani HTML kod neovisno o platformi i prikazuju na ekranu konačni oblik stranica. Dobro poznavanje HTML jezika je pretpostavka za kvalitetno dizajniranje web stranica. Osim oznaka s njihovim atributima, vrlo značajan dio HTML jezika su obrasci (engl. *forms*) pomoću kojih se unose podaci na web stranice. Takav unos podataka omogućava autorizaciju i unos u baze podataka za koje nije odgovoran HTML već druge tehnologije koje će biti opisane u nastavku.

Prvi dokument koji opisuje HTML izdan je 1990. godine i zvao se „HTML Tags“, a autor je bio Tim Berners-Lee. Od tada se HTML stalno razvija. Od 1996. godine HTML specifikacije održava World Wide Web Consortium (W3C) iako je 2000. godine HTML postao međunarodni standard (ISO/IEC 15445:2000).

HTML4 predstavljen je u prosincu 1997. godine. Nastavio je s prihvaćanjem oznaka nametnutih od strane proizvođača različitih web preglednika, no istovremeno je pokrenuto i "čišćenje" standarda proglašavanjem nekih od njih suvišnim.

HTML 5 donosi brojne nove mogućnosti koje HTML 4.01 i XHTML 1.x nisu imali, kao što je mogućnost reprodukcije videa na stranicama bez korištenja *Adobe flash playera*, mogućnost upravljanja pomoću tipkovnice i opcijama za bilo koju vrstu manipulacija, *drag and drop, canvas* kao i ostalih novih elemenata.

Inačica korištena u izradi završnog rada je HTML5.

2.2. Javascript

Javascript je najpopularniji jezik za programiranje na webu. Izumio ga je Brendan Eich iz tvrtke Netscape i pojavljuje se u svim preglednicima od 1996. godine. Službeno ga je standardizirala 1997. godine ECMA organizacija (European Computer Manufacturers Association), a 1998. godine je odobren kao međunarodni ISO (ISO/IEC 16262) standard.

Javascript omogućuje web stranici interakciju s korisnikom. Na primjer tekst koji se mijenja ovisno o akcijama korisnika, obrada podataka koje je korisnik unio, mogućnost detekcije korištenog web preglednika i prosljeđivanje valjane stranice na pregled i sl.

Javascript se izvodi na računalu korisnika (engl. *client side*) i zbog sigurnosnih razloga ima neka ograničenja (npr. nemoguće je spremiti datoteku).

2.3. PHP

PHP je otvoreni (engl. *open source*) skriptni jezik za dinamičko generiranje HTML koda koji se izvodi na strani poslužitelja (engl. *server side*). Zapravo to je skraćenica od "PHP: Hypertext Preprocesor". Drugim riječima, PHP je skriptni jezik pomoću kojeg se mogu kreirati HTML stranice na poslužitelju prije nego što se iste, popunjene dinamičkim sadržajem, pošalju klijentu. Ovim načinom generiranja sadržaja klijent ne može vidjeti kod (skriptu) koji je generirao sadržaj koji gleda, već ima pristup samo čistom generiranom HTML kodu.

Vrlo jednostavno, PHP je jedan od najpopularnijih i najmoćnijih skriptnih jezika trenutno na tržištu. Broj stranica koji koriste PHP raste iz dana u dan, a broj tvrtki koje žele primijeniti PHP na svojim stranicama je još veći. PHP je izvrstan jer se pomoću njega s nevjerojatnom lakoćom mogu stvoriti opširne web aplikacija s velikim količinama podataka.

Otvoreni u gornjoj definiciji znači da svatko tko želi može skinuti izvorne PHP kodove pisane u C-u i, ukoliko ih razumije, može ih mijenjati po svojoj volji te dodavati nove funkcije PHP-u. Štoviše, svi su pozvani da sudjeluju u razvoju novih inačica PHP-a. Izvorni kodovi i instalacijske datoteke mogu se skinuti sa službenih PHP stranica (http://php.net).

Ono što PHP stavlja još više ispred ostalih web skriptnih tehnologija je njegova podrška za upravljanje širokom paletom baza podataka. Podržava sve popularnije baze podataka kao što su MySQL, PostgreSQL, dBase, Oracle, ODBC, MSSQL i dr.

Inačica korištena u izradi završnog rada je PHP 5.62.

2.4. CSS

Izvornog naziva Cascade Style Sheets predstavlja „stilski" jezik koji se najčešće koristi za opis prezentacijske sheme HTML dokumenta, ali može se koristiti za bilo koju inačicu XML (EXtensible Markup Language*)* dokumenata. Osmišljen je kako bi se odvojio sadržaj od vizualnog dijela uključujući izgled, boje i fontove, te kako bi daljnje promjene bile jednostavnije jer jedna

CSS datoteka može biti povezana s više datoteka sa sadržajem. Također sadržaj se može različito prikazivati ovisni o metodi iscrtavanja (na ekranu ili na papiru).

Inačica korištena u izradi završnog rada je CSS3.

2.5. jQuery

Najpopularnija *javascript* skripta na webu koju koristi 65% od 10 milijuna najposjećenijih web stranica na webu. Objavio ju je u siječnju 2006. godine John Resig kao softver otvorenog koda. Svrha ove skripte je pojednostavljeno kodiranje na klijentskoj strani, lakši pristup DOM (Document Object Model) elementima, upravljanje događajima, a omogućava i korištenje lijepih animacija te razvoj AJAX aplikacija. Modularni pristup jQuery biblioteka omogućuje stvaranje moćnih dinamičkih web stranica i web aplikacija. jQuery, je u osnovi DOM (Document Object Model) upravljačka biblioteka. DOM je prikaz strukturnog grananja svih elemenata web stranice i jQuery pojednostavljuje sintaksu za traženje, odabir i upravljanje tim DOM elemenata. Na primjer, jQuery se može koristiti za pronalaženje elementa u dokumentu s određenim svojstvima (npr. svi elementi označeni sa h1 oznakom), promjenu jednog ili više svojih atributa (npr. boja, vidljivosti), ili odgovoriti na događaj (npr. klik miša).

2.6. XAMPP

Besplatni web poslužitelj otvorenog koda koji je moguće koristiti na različitim platformama (Linux, Windows, MasOS, Solaris), a razvijen kroz neprofitabilni projekt Apache Friends. Sastoji se od Apache web poslužitelja te MySQL baze podataka koja je 19. listopada 2015. godine zamijenjena MariaDB bazom podataka, a ima mogućnost interpretirati skripte napisane u PHP-u i Pearl-u. Prvotno namijenjen samo kao pomagalo programerima prilikom razvoja ali od kada mnogi komercijalni poslužitelji koriste ista rješenja kao i XAMPP, prebacivanje aplikacija sa razvojne u produkcijsku okolinu postaje iznimno lako. Također uz XAMPP dolaze i drugi moduli kao što su OpenSSL, phpMyAdmin, MediaWiki, Joomla, Wordpress i dr. Moguća je koegzistencija više instanci na jednom računalu. Isporučuje se u potpunoj i standardnoj (manjoj) inačici.

2.7. Microsoft SQL

Microsoft SQL server je relacijska baza podataka. Proizvod je razvila tvrtka Microsoft nakon što je izvorno od tvrtke Sybase preuzela licenčna prava i izvorni kod. Izvorno je proizvod nosio

naziv Sybase SQL server i bio je namijenjen uporabi na OS/2 platformi. Slijedeća zajednička inačica proizvoda nosila je ime „SQL Server for OS/2 1.0“ koju je izdao Microsoft, dok je identičnu inačicu izdao i Sybase pod nazivom „SQL Server 3.0“. Godine 1992. Microsoft izdaje posljednju inačicu pod nazivom Microsoft SQL Server 4.2 koja se još uvijek pokretala na OS/2 platformi. Nakon preuzimanja prava i razvojem NT platforme Microsoft je usmjerio razvoj SQL relacijske baze podataka prema Windows NT platformi s idejom da baza postane usko povezana sa NT operacijskim sustavom. Bitne razlike u razvoju nastale su nakon potpunog odvajanja ove dvije tvrtke. Nova samostalna inačica Microsoft SQL servera imala mogućnost grafičkog pristupa upravljanju podacima, odnosno grafičko korisničko sučelje, što je bilo jedinstveno u svijetu. To je znatno olakšalo upravljanje strukturom podataka, kao i samim podacima. Do tada se koristio tzv. „command line shell“ odnosno tekstualno sučelje koje je bilo vrlo zahtjevno, a istovremeno nepregledno.

Na toj osnovi u svim je daljnjim inačicama usavršavano korisničko sučelje, ali i sve druge tehnologije razvijane paralelno sa razvojem osnovnih komponenti. Značajan napredak dogodio se razvojem inačica 2000 i 2005 gdje su realizirana mnoga rješenja među kojima su: SQL Server Integration Services, OLAP server, Notification services, Reporting Server i svakako klijentski dio IDE sučelja.

Microsoft SQL server temelji se na Transact SQL jeziku koji potiče još od suradnje sa Sybase-om. Glavna karakteristika jezika je da osim osnovnih SQL upita (SELECT, INSERT, DELETE …) ima mogućnost upravljanja tokom programa i sadrži naredbe za petlje (IF, FOR..NEXT, WHILE …). Ta značajka je izuzetno bitna u razvoju aplikacija koje se temelje na relacijskim bazama jer se kroz store procedure i funkcije može manipulirati podacima i kvalitetno ih pripremiti za izvješća. Na taj se način bitno povećava brzina dohvata željenih podataka.

Višeplatformski početak razvoja pretpostavio je potrebu da se zadrži mogućnost pristupa Microsoft SQL Serveru sa drugih platformi, pa su za te potrebe razvijeni različiti protokoli poput FreeTDS (Free Tabular Data Stream) protokola i ODBC (Open Database Connectivity) protokola.

Daljnjim razvojem došlo je do potrebe implementacije web servisa, pa se u kasnijim inačicama (od 2005 pa dalje) nalaze SOAP W3C (Service Oriented Architecture Protocol), JDBC API (sučelje za komunikaciju sa Java aplikacijama) i dr.

Razvijen je sigurnosni mehanizam zrcaljenja (engl. *mirroring*) i sustav grupiranja (engl. *clustering*) koji će biti opisani u daljnjem tekstu.

Inačica korištena u izradi završnog rada je Microsoft SQL Server 2008 R2.

3. RAZVOJ APLIKACIJE

3.1. Knjigovodstvene evidencije i praćenja

Menadžerske odluke pretpostavljaju kvalitetno, točno i ažurno knjigovodstvo s podacima u realnom vremenu i zbog toga je potreba pristupa izvješćima danas neophodna. Knjigovodstvo tvrtke, koja je obveznik poreza na dohodak, vodi se na principima dvojnog knjigovodstva.

Dvojno knjigovodstvo propisuje vođenje *salda-konti* analitičke evidencije kupaca i dobavljača, te vođenje glavne knjige odnosno financijskog knjigovodstva.

Salda-konti kupaca i dobavljača je kronološka evidencija izlaznih i ulaznih računa te uplata po njima. Na osnovu tih podataka iz baze se mogu kreirati kartice za svakog pojedinog partnera i skupna izvješća kao što su popisi otvorenih stavki, odnosno neplaćenih računa.

Financijsko knjigovodstvo kronološki evidentira sve financijske promjene po određenom kontnom (računskom) planu i na osnovu tih podataka kreiraju se izvješća poput bilance, financijskih kartica, analitičkih i podanalitičkih izvješća, pregleda prihoda i rashoda, obveza i troškova te na kraju poslovnog rezultata u određenim vremenskim razdobljima.

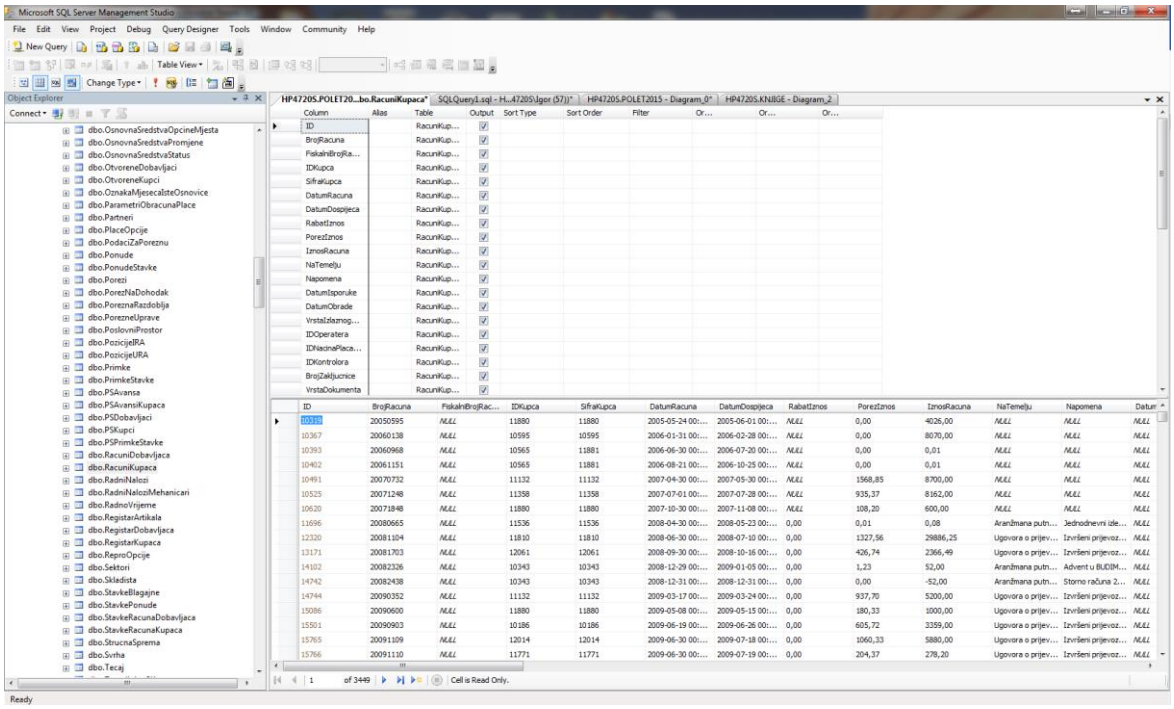
Nasuprot dvojnom knjigovodstvu, obrti i tvrtke koje su obveznici poreza na dohodak vode jednostavno knjigovodstvo, pa ova aplikacija nije namijenjena njima jer jedan bitan dio knjigovodstva ne vode pa u aplikaciji kreirana izvješća nemaju dostupne podatke.

3.2. Baza podataka

Baza podataka izrađena je na MS SQL Serveru inačice 2008 R2. Za potrebe izrade, kao i izvođenja aplikacije potrebno je instalirati SQL express inačicu koja je besplatna i dovoljna unatoč svojim ograničenjima. Baza se sastoji od nešto više od 130 tablica i 230 *store* procedura i kao takva podloga je za kompletno knjigovodstveno rješenje jednog poduzeća. Za potrebe ove aplikacije korišten je jedan manji dio navedenih resursa, tj. onaj dio koji se odnosi na financijska praćenja trenutnih stanja i jedna baza koja se sastoji od samo jedne tablice i služi u autorizacijske svrhe prilikom prijave u aplikaciju.

Kreiranje tablica, *store* procedura, postavljanje relacijskih veza između tablica, kreiranje indeksa, pogleda i sl. obavlja se u grafičkom korisničkom sučelju SQL poslužitelja pod nazivom *Microsoft SQL Server Management Studio* koji je sastavni dio svih inačica SQL servera poslije inačice SQL 2005. (Sl. 3.3.). Ovaj je alat od izuzetne pomoći u kreiranju, organizaciji i održavanju

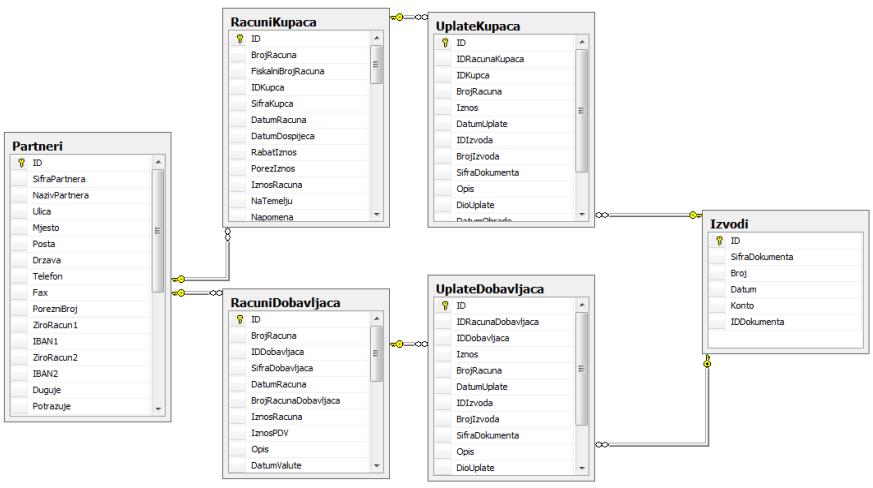
baza podataka jer je sve vizualno predočeno nasuprot *command line* alatima koji su korišteni u prvim inačicama SQL Servera.



Slika 3.1. Microsoft SQL Server Management Studio

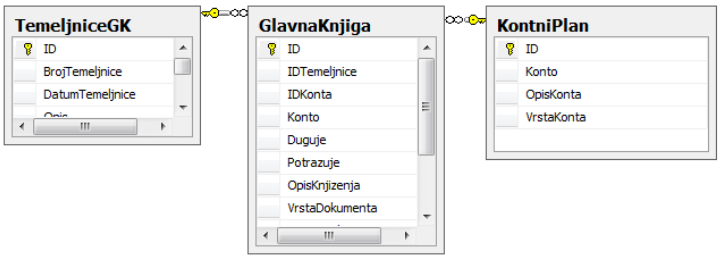
Prijava u aplikaciju standardno koristi podatke o korisničkom imenu i pripadajućoj zaporki, te na osnovu ta dva podatka aplikacija vrši izbor baze podatka određene tvrtke. U ovom radu korištena je samo jedna baza sa podacima.

Za kartice kupaca i dobavljača te otvorene stavke korištene su tablice *RacuniKupaca*, *UplateKupaca*, *RacuniDobavljaca*, *UplateDobavljaca* te tablice u relacijskoj vezi *Partneri* i *Izvodi* (Sl. 3.2.).



Slika 3.2. Tablice u relacijskoj vezi na osnovu kojih su izrađene kartice kupaca i dobavljača

Za financijska izvješća korištena je tablica *GlavnaKnjiga* te tablice u relacijskoj vezi *KontniPlan* i *TemeljniceGK* (Sl. 3.3.).



Slika 3.3. Tablice u relacijskoj vezi na osnovu kojih su izrađena financijska izvješća

Izbor partnera odnosno kupca ili dobavljača kreiran je SELECT upitom iz koda:

$partneri = "SELECT NazivPartnera, CAST(SifraPartnera AS nvarchar(5)) AS Sifra FROM Partneri WHERE SifraPartnera IN (SELECT SifraKupca FROM RacuniKupaca) ORDER BY NazivPartnera";

za kupce, odnosno:

$partneri = "SELECT NazivPartnera, CAST(SifraPartnera AS nvarchar(5)) AS Sifra FROM Partneri WHERE SifraPartnera IN (SELECT SifraDobavljaca FROM RacuniDobavljaca) ORDER BY NazivPartnera";

za dobavljače. Iz priloženih primjera koda vidljivo je pozivanje polja naziv i šifra partnera koji se prosljeđuju *store* procedurama.

Store procedure kreirane su u bazi i sastavni su dio iste, a izgledaju kao na slici 3.4.

Sve korištene procedure nalaze se u prilogu.

```
/****** Object:  StoredProcedure [dbo].[KarticaDobavljacaPHP]    Script Date: 06/19/2016 23:19:04 ******/
SET ANSI_NULLS OFF
GO
SET QUOTED_IDENTIFIER OFF
GO

CREATE PROCEDURE [dbo].[KarticaDobavljacaPHP] @sifdob INT AS
SELECT     TOP 100 PERCENT MIN(dbo.RacuniDobavljaca.ID) AS RacunID, dbo.RacuniDobavljaca.SifraDobavljaca,

dbo.RacuniDobavljaca.IDDobavljaca, dbo.RacuniDobavljaca.BrojRacuna,
                      CAST(SUM(dbo.RacuniDobavljaca.IznosRacuna) AS NVARCHAR(15)) AS IznosRacuna, SUM

(dbo.RacuniDobavljaca.IznosPDV) AS IznosPDV, CAST(ISNULL(sud.DugujeD, 0.00) AS NVARCHAR(15))
                      AS IznosUplate,  MIN(dbo.RacuniDobavljaca.BrojRacunaDobavljaca) AS BrojRacunaDobavljaca,
                      MIN(dbo.RacuniDobavljaca.PozivNaBroj2) AS PozivNaBroj2, CONVERT(NVARCHAR(10),MIN

(dbo.RacuniDobavljaca.DatumRacuna),104) AS DatumRacuna, CONVERT(NVARCHAR(10),MIN(dbo.RacuniDobavljaca.DatumValute),104) AS

DatumValute,
                      CAST((ISNULL(SUM(dbo.RacuniDobavljaca.IznosRacuna) - ISNULL(sud.DugujeD, 0.00), 0.00)) AS NVARCHAR

(15)) AS Saldo
FROM         dbo.RacuniDobavljaca LEFT OUTER JOIN
                         (SELECT     IDDobavljaca, BrojRacuna, SUM(Iznos) AS DugujeD
                           FROM          dbo.UplateDobavljaca
                           GROUP BY IDDobavljaca, BrojRacuna) sud ON dbo.RacuniDobavljaca.BrojRacuna = sud.BrojRacuna
GROUP BY dbo.RacuniDobavljaca.BrojRacuna, dbo.RacuniDobavljaca.SifraDobavljaca, dbo.RacuniDobavljaca.IDDobavljaca,

sud.DugujeD
HAVING      (dbo.RacuniDobavljaca.SifraDobavljaca like @sifdob)
ORDER BY dbo.RacuniDobavljaca.BrojRacuna
```

Slika 3.4. Store procedura KarticaDobavljacaPHP koja dohvaća podatke iz baze.

Iz gornjih navoda vidljivo je da je moguće direktno iz aplikacije pozivati SELECT upite, odnosno bilo koji T-SQL upit, te isto tako slati samo parametre postojećim upitima. U pravilu većina se upita kreira u samoj bazi, a iznimno, jednostavniji upiti se mogu kreirati i u PHP kodu.

Po istim principima kreirane su *store* procedure za otvorene stavke, financijsku karticu, bilancu, podanalitike i preglede prihoda i rashoda.

Ovakvim upitima u bazu podataka dobivaju se slike (engl. *snapshots*) trenutnih podataka u bazi. Tako je moguće u kratkom vremenu uočiti promjene ako se knjigovodstvo vodi u realnom vremenu.

Fizički, svi se podaci nalaze na jednom poslužitelju u tvrtki, a tehnologijama poput *mirroringa* i *clusteringa* moguće je da se isti nalaze i na više poslužitelja na različitim geografskim lokacijama, ali ipak logički povezani kao jedna cjelina.

Mirroring odnosno zrcaljenje omogućava preslikavanje svih podataka na najmanje još jednu lokaciju, pa se u slučaju nedostupnosti jednog od poslužitelja pristupa drugom i osigurava se nesmetani rad aplikacije.

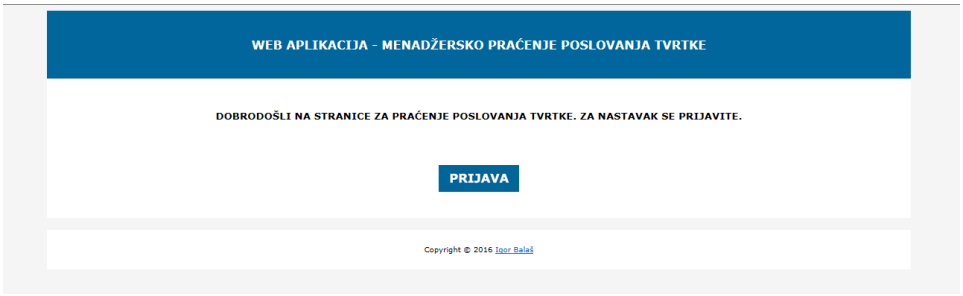
Clustering odnosno grupiranje tehnologija koja omogućava zapis pojedinih dijelova jedne baze podataka na više poslužitelja i koristi se kod velikih baza koje kapacitetima prelaze mogućnosti jednog poslužitelja.

3.3. Sučelje

Sučelje aplikacije napravljeno je pomoću u HTML5 jezika za prikaz sadržaja, CSS3 opisnog prezentacijskog jezika, PHP-a, Javascripta i jQuery-a.

HTML je jezik kojeg internetski preglednici interpretiraju i na osnovu toga prikazuju sadržaj na ekranu. U svakom pregledniku postoji opcija pregleda izvornog koda, što je ustvari čisti HTML kod i ništa drugo od navedenih tehnologija nije vidljivo.

Početna slika aplikacije je interpretiranje datoteke *index.php* i izgleda kao na slici 3.5.



Slika 3.5. Početna strana aplikacije – index.php

Kod početne stranice izgleda ovako:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <meta http-equiv="Content-Language" content="hr" />
<title>Web aplikacija - menadžersko praćenje poslovanja</title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
<style type="text/css">
body {
  background-color: #F5F5F5;
}
</style>
</head>
<body>
<div class="container">
 <div class="header">
   <p>WEB APLIKACIJA - MENADŽERSKO PRAĆENJE POSLOVANJA TVRTKE</p>
 </div>
     
 <h5 align="center">DOBRODOŠLI NA STRANICE ZA PRAĆENJE POSLOVANJA TVRTKE. ZA NASTAVAK SE PRIJAVITE.</h5>
       
 <div align="center">
   <p><a href="prijava.php" target="_parent"><img src="PRIJAVA.gif" width="108" height="36"/></a>
   </p>
   <p> </p>
 </div>
 <div class="footer">
   <p>
Copyright &copy; 2016 <a href="mailto:ibalas@etfos.hr">Igor Bala&scaron;</a>
</p>
</div>
<div align="center"></div>
</div>
</body>
</html>
```

Iz koda se vidi čisti HTML kod koji poziva proceduru *prijava.php*. Ono što se ne vidi u kodu, a daje dizajn stranici sadržano je u datoteci *styles.css* koja je uključena u *<head>* oznaci. Već ranije je objašnjena uloga CSS-a pa ovdje prilažemo dio koda iz *styles.css* datoteke koji se

odnosi na datoteku *index.php*. To su dijelovi koji su pozvani atributima *class="container", class="header"* i *class="footer"* navedenima unutar <*div*> oznaka.

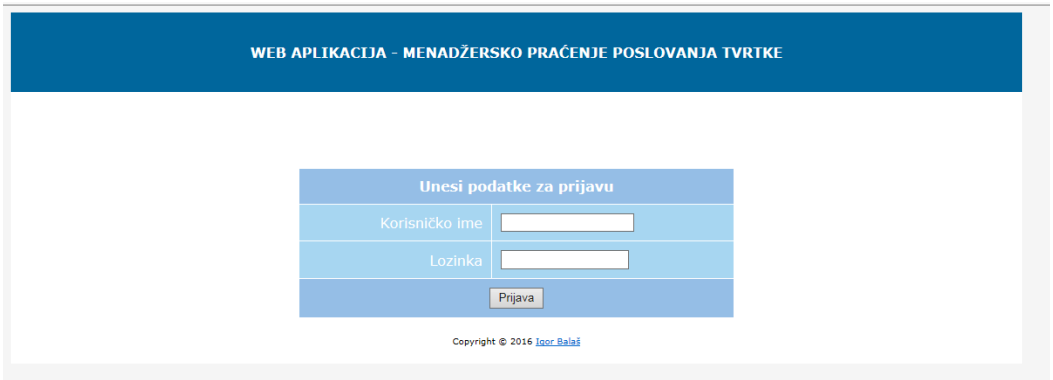
```
@charset "utf-8";
.container {
  height: auto;
  width: 1160px;
  margin-right: auto;
  margin-left: auto;
  background-color:#FFF;
}
.header {
  height: auto;
  width: 1160px;
  padding-top: 20px;
  padding-bottom: 20px;
  margin-right: auto;
  margin-left: auto;
  background-color: #00669c;
  font-style: normal;
  color: #FFF;
  vertical-align:center;
  text-align:center;
  font-weight: bold;
}
.footer {
  height: auto;
  width: 1140px;
  margin-right: auto;
  margin-left: auto;
  text-align: center;
  float: left;
  font-size: x-small;
  padding: 10px;
  background-color:#FFF;
}
```

Iz navedenog koda vidljivo je da se unutar <*div*> oznaka označenih sa *class*="nešto" pozivaju podaci iz CSS datoteke koji HTML-u služe za interpretaciju stranice, kao što su širina i visina oznaka, margine, tekstualna poravnanja, veličine fonta, boje pozadine, boje fonta i dr.

Na ovaj način omogućeno je da se u svim datotekama koje tvore aplikaciju mogu pozivati jednaki parametri samo uključivanjem CSS datoteke u <*head*> oznaci i definiranjem atributa *class*

uz pojedinu <*div*> oznaku. U suprotnome, bilo bi potrebno u svakoj datoteci navoditi sve parametre za prikaz. Za to je zaslužna CSS tehnologija. Eventualne promjene u dizajnu vrlo se jednostavno realiziraju promjenom na samo jednom mjestu u CSS datoteci, a reflektiraju se u svim datotekama aplikacije.

Klikom na ikonu prijava otvara se stranica *prijava.php* (Sl. 3.6). Stranica sadrži čisti HTML kod i služi za autorizaciju korisnika aplikacije, odnosno za unos podataka o korisniku. Nakon klika na gumb Prijava poziva se slijedeća datoteka *login.php*.



Slika 3.6. Stranica za autorizaciju – prijava.php

Datoteka *login.php* je čista php datoteka koja se izvršava na poslužiteljskoj strani i nema nikakve interakcije sa korisničkim sučeljem. Izvorni kod izgleda ovako:

```
<?php
session_start();
$message="";
if(count($_POST)>0)
{
/* Specify the server and connection string attributes. */
$serverName =file_get_contents("server.txt");

/* Get UID and PWD from application-specific files.  */
$uid = file_get_contents("uid.txt");
$pwd = file_get_contents("pwd.txt");
$dbname = file_get_contents("baza.txt");
$connectionInfo = array("UID"=>$uid,"PWD"=>$pwd,"Database"=>$dbname, "CharacterSet" => "UTF-8");

/* Connect using SQL Server Authentication. */
$connw = sqlsrv_connect( $serverName, $connectionInfo);
if( $connw === false )
{
    echo "Unable to connect.</br>";
    die( print_r( sqlsrv_errors(), true));
```

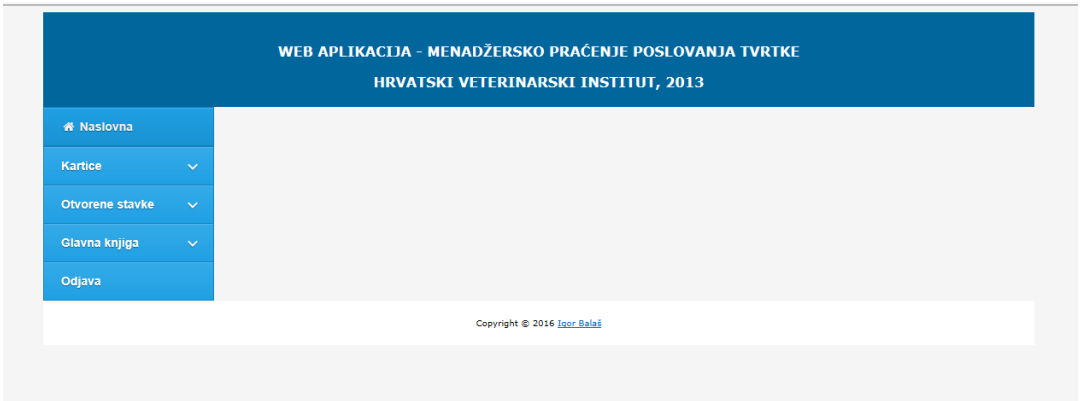
```
};

$korid ="SELECT Ime, BazaNaziv, BazaUsername, BazaPassword, BazaServer FROM korisnici WHERE (username='" .
$_POST["username"] . "') AND (password = '". $_POST["password"]."')";
$korid_query = sqlsrv_query($connw, $korid, array(), array( "Scrollable" => 'keyset' ));
$row = sqlsrv_fetch_array($korid_query,SQLSRV_FETCH_BOTH);

if(is_array($row)) {
$_SESSION["ime"] = $row['Ime'];
$_SESSION["bazaname"] = $row['BazaNaziv'];
$_SESSION["bazausername"] = $row['BazaUsername'];
$_SESSION["bazapass"] = $row['BazaPassword'];
$_SESSION["bazaserver"] = $row['BazaServer'];
if(isset($_SESSION["ime"])) {
header("Location:naslovna.php");
}
} else {
$message = "Invalid Username or Password!";
header("Location:prijava.php");
}
}
?>
```

U priloženom kodu se vidi da je sav kod unutar <?PHP ?> oznake i govori o tome da je cijela datoteka ustvari PHP skripta. Kod preuzima unesene podatke iz prethodne datoteke i poziva jedan SELECT upit prema autorizacijskoj bazi podataka. Upit se kreira na osnovu podataka o nazivu poslužitelja, nazivu baze, korisničkom imenu i zaporki koje su pročitane iz konfiguracijskih datoteka (*server.txt, uid.txt, pwd.txt* i *baza.txt*). U slučaju da su uneseni podaci ispravni kreiraju se $_SESSION varijable koje se prenose na cijelu aplikaciju i poziva se slijedeća datoteka *naslovna.php*. U suprotnom aplikacija ponovo traži unos ispravnih podataka. $_SESSION varijable nose podatke o nazivu poslužitelja, nazivu baze, korisničkom imenu i zaporki knjigovodstvene baze podataka i koriste se u svim izvješćima u aplikaciji.

Stranica *naslovna.php* izgleda kao na slici 3.7 i sadrži izbornik za izbor izvješća te u naslovnoj traci ime tvrtke na koju se podaci odnose.



Slika 3.7. Naslovna stranica sa izbornikom – naslovna.php

S ove stranice moguće je kroz izbornik odabrati koje izvješće želimo. To je stranica koja sadrži u svom izvornom kodu HTML i PHP, a poziva Javascript i jQuery. Kod izgleda ovako:

```
<?php
include('login.php');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <meta http-equiv="Content-Language" content="hr" />
<title>Web aplikacija - menadžersko praćenje poslovanja</title>
  <script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="css/style.css">
<style type="text/css">
body {
  background-color: #F5F5F5;
}
</style>
</head>

<body>
<?php

/* Specify the server and connection string attributes. */
//$serverName =file_get_contents("server.txt");

/* Get UID and PWD from application-specific files.  */
$uidb = $_SESSION["bazausername"];
$pwdb = $_SESSION["bazapass"];
```

```
$baz = $_SESSION["bazaname"];
$serverName = $_SESSION["bazaserver"];
$connectionInfo = array("UID"=>$uidb,"PWD"=>$pwdb,"Database"=>$baz, "CharacterSet" => "UTF-8");


//$connectionInfo = array(file_get_contents("d:/stranica/veza.txt"));

/* Connect using SQL Server Authentication. */
$conn = sqlsrv_connect( $serverName, $connectionInfo);
if( $conn === false )
{
    echo "Unable to connect.</br>";
    die( print_r( sqlsrv_errors(), true));
};
$tvrtka = "SELECT NazivFirme, PoslovnaGodina FROM Tvrtka";
//izvrsavanje querya za dogadjaje
$tvrtka_query = sqlsrv_query($conn, $tvrtka, array(), array( "Scrollable" => 'keyset' ));
?>
<div class="container">
  <div class="header">
    <p>WEB APLIKACIJA - MENADŽERSKO PRAĆENJE POSLOVANJA TVRTKE</p>
<?PHP
  for ($count = 1; $rowt = sqlsrv_fetch_array($tvrtka_query,SQLSRV_FETCH_BOTH); ++$count)
      {
  echo ("$rowt[0]");
  echo (", ");
  echo ("$rowt[1]");
      };
?>
   </div>
<div id="cssmenu">
  <ul>
     <li class="active"><a href="naslovna.php"><span><i class="fa fa-fw fa-home"></i> Naslovna</span></a></li>
      <li class="has-sub"><a href="#"><span>Kartice</span></a>
        <ul>
          <li><a href="karticakupca.php"><span>Kupci</span></a></li>
          <li><a href="karticadobavljaca.php"><span>Dobavljaci</span></a></li>
        </ul>
       </li>
     <li class="has-sub"><a href="#"><span>Otvorene stavke</span></a>
         <ul>
          <li><a href="otvorenekupci.php"><span>Kupci</span></a></li>
          <li><a href="otvorenedobavljaci.php"><span>Dobavljaci</span></a></li>
         </ul>
      </li>
```

```
    <li class="has-sub"><a href="#"><span>Glavna knjiga</span></a>
       <ul>
         <li><a href="financijskakartica.php"><span>Financijska kartica</span></a></li>
         <li><a href="bilanca.php"><span>Bilanca</span></a></li>
         <li><a href="podanalitike.php"><span>Podanalitke</span></a></li>
         <li><a href="prihodirashodi.php"><span>Prihodi i rashodi</span></a></li>
       </ul>
    </li>
    <li><a href="logout.php"><span>Odjava</span></a></li>
</ul>
</div>


<div class="footer">
<p>
Copyright &copy; 2016 <a href="mailto:ibalas@etfos.hr">Igor Bala&scaron;</a>
</p>
</div>
</div>

</body>
</html>
```

Vidljivo je da kod počinje *include('login.php')* naredbom koja je potrebna kako bi se preuzele važeće $_SESSION varijable kako bi se mogle koristiti za spajanje na knjigovodstvenu bazu podataka.

Izbornik je napravljen pomoću Javascipta i jQuery-a. Sam izgled izbornika definiran je HTML kodom i pripadajućom CSS datotekom. Međutim, sve akcije koje se događaju kada se klika po izborniku definirane su u *script.js* datoteci čiji kod izgleda ovako:

```
(function($){
$(document).ready(function(){

$('#cssmenu li.active').addClass('open').children('ul').show();
$('#cssmenu li.has-sub>a').on('click', function(){
  $(this).removeAttr('href');
  var element = $(this).parent('li');
  if (element.hasClass('open')) {
    element.removeClass('open');
    element.find('li').removeClass('open');
    element.find('ul').slideUp();
  }
  else {
    element.addClass('open');
```

```
    element.children('ul').slideDown();
    element.siblings('li').children('ul').slideUp();
    element.siblings('li').removeClass('open');
    element.siblings('li').find('li').removeClass('open');
    element.siblings('li').find('ul').slideUp();
  }
});

$('#cssmenu>ul>li.has-sub>a').append('<span class="holder"></span>');

(function getColor() {
  var r, g, b;
  var textColor = $('#cssmenu').css('color');
  textColor = textColor.slice(4);
  r = textColor.slice(0, textColor.indexOf(','));
  textColor = textColor.slice(textColor.indexOf(' ') + 1);
  g = textColor.slice(0, textColor.indexOf(','));
  textColor = textColor.slice(textColor.indexOf(' ') + 1);
  b = textColor.slice(0, textColor.indexOf(')'));
  var l = rgbToHsl(r, g, b);
  if (l > 0.7) {
    $('#cssmenu>ul>li>a').css('text-shadow', '0 1px 1px rgba(0, 0, 0, .35)');
    $('#cssmenu>ul>li>a>span').css('border-color', 'rgba(0, 0, 0, .35)');
  }
  else
  {
    $('#cssmenu>ul>li>a').css('text-shadow', '0 1px 0 rgba(255, 255, 255, .35)');
    $('#cssmenu>ul>li>a>span').css('border-color', 'rgba(255, 255, 255, .35)');
  }
})();

function rgbToHsl(r, g, b) {
    r /= 255, g /= 255, b /= 255;
    var max = Math.max(r, g, b), min = Math.min(r, g, b);
    var h, s, l = (max + min) / 2;

    if(max == min){
        h = s = 0;
    }
    else {
        var d = max - min;
        s = l > 0.5 ? d / (2 - max - min) : d / (max + min);
        switch(max){
            case r: h = (g - b) / d + (g < b ? 6 : 0); break;
            case g: h = (b - r) / d + 2; break;
```

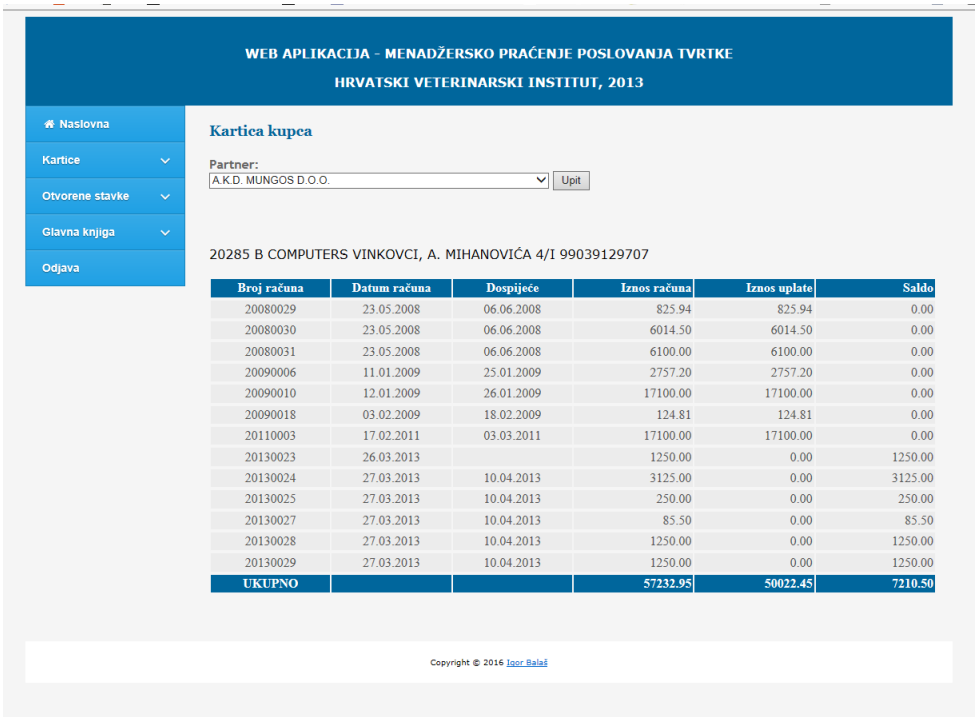
```
            case b: h = (r - g) / d + 4; break;
        }
        h /= 6;
    }
    return l;
}

});
})(jQuery);
```

Ova datoteka izvodi se na klijentskoj stranici i izvođenje bilo kojeg njezinog dijela odmah je vidljivo u pregledniku.

Izbor određene stavke menija pokreće datoteku navedenu u *href* atributu u listama <*li*> definiranog izbornika.

Stranice kartica kupca i kartica dobavljača vizualno su iste i za njih je karakterističan SELECT upit prema bazi za kreiranje padajućeg izbornika partnera. Kartica izgleda kao na slici 3.8.



Slika 3.7. Kartica kupca

Na slici je vidljiv padajući izbornik koji se puni pozivom upita:

$partneri = "SELECT NazivPartnera, CAST(SifraPartnera AS nvarchar(5)) AS Sifra FROM Partneri WHERE SifraPartnera IN (SELECT SifraKupca FROM RacuniKupaca) ORDER BY NazivPartnera";

i nakon klika na gumb *Upit* aplikacija vrši upit za odabranog partnera pozivanjem *store* procedure iz PHP koda:

```
$sifdob = $_GET['select'];

    $params = array(array($sifdob, SQLSRV_PARAM_IN));
    $storka="{call dbo.KarticaKupcaPHP(?)}";
    $result = sqlsrv_query($conn, $storka, $params);
```

i prikazuje tablicu sa dobivenim rezultatima:

```
for ($count = 1; $row = sqlsrv_fetch_array($result,SQLSRV_FETCH_BOTH); ++$count)
    {
            echo("<table width=\"910\" border=\"0\" cellspacing=\"2\" cellpadding=\"2\">");
            echo("<tr><td width=\"130\" align=\"center\" valign=\"center\">");
            echo ("$row[3]");
            echo("</td><td width=\"130\" align=\"center\" valign=\"center\">");
            echo ("$row[7]");
            echo("</td><td width=\"130\" align=\"center\" valign=\"center\">");
            echo ("$row[8]");
            echo("</td><td width=\"130\" align=\"right\" valign=\"center\">");
            echo ("$row[4]");
            echo("</td><td width=\"130\" align=\"right\" valign=\"center\">");
            echo ("$row[6]");
            echo("</td><td width=\"130\" align=\"right\" valign=\"center\">");
            echo ("$row[9]");
            echo ("</td></tr></table>");}
```

Na istom principu rade kartica dobavljača i financijska kartica.

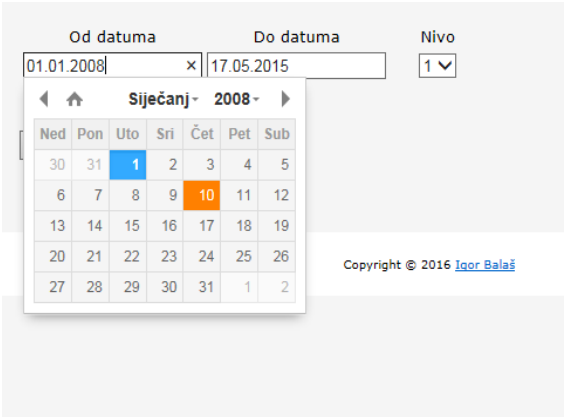
Izvješća otvorene stavke, bilanca, pregledi prihoda i rashoda i podanalitike koriste *datetime picker* za izbor željenog datuma koji označava period izvješća.

Stranica *bilanca.php* izgleda kao na slici 3.9.

Slika 3.9. Stranica bilanca – bilanca.php

Datumi od datuma do datuma se mogu unijeti ručno ili pomoću *datetime pickera*, a označavaju period izvješća, padajući izbornik *Nivo* sadrži vrijednosti od 1 do 8 i označava koliko znamenki konta izvješće koristi pri kreiranju. U primjeru na slici korišten je nivo 1, odnosno zbrojeni su svi podaci prema prvoj znamenci konta.

Datetime picker kreiran je pomoću Javascripta i izgleda kao na slici 3.10.



Slika 3.10. Datetime picker – izbornik datuma

Klikom na odabir Odjava u izborniku pokreće se stranica *logout.php*. Radi se o čistoj PHP stranici koja svojom naredbom *unset* briše $_SESSION varijable i poziva početnu *index.php* stranicu. Kod izgled ovako:

```
<?php
session_start();
```

```
unset($_SESSION["username"]);
unset($_SESSION["password"]);
unset($_SESSION["ime"]);
unset($_SESSION["bazaname"]);
unset($_SESSION["bazausername"]);
unset($_SESSION["bazapass"]);
header("Location:index.php");
?>
```

4. ZAKLJUČAK

Prilikom realizacije ovog rada kreirana je aplikacija za praćenje poslovanja što će omogućiti menadžerima donošenje kvalitetnijih poslovnih odluka. Razvoj ove aplikacije zaokružuje cjelinu knjigovodstvenog rješenja većih tvrtki

Aplikacija prikazuje izvješća identična onima koja koristi knjigovodstvena služba na svojim desktop aplikacijama. U karticama kupaca i dobavljača u padajućim izbornicima su ponuđeni samo oni partneri koji uistinu imaju nekakve evidencije kako bi izbacili sve nepotrebne podatke koji putuju mrežom i na taj način maksimalno povećali brzinu protoka podataka. Kod izvješća koja se kreiraju u određenom vremenskom periodu ponuđeni datumi su kreirani na osnovu upita o najmanjem i najvećem datumu koji se pojavljuje unutar evidencije. Na taj način je pojednostavljen unos željenih parametara i povećana jednostavnost uporabe.

Svi elementi aplikacije su isključivo izvještajni, dakle nema unosa u bazu podataka, što niti nije funkcija menadžera u tvrtki.

Ispisi pojedinih izvješća nisu izrađeni u sklopu aplikacije jer nije za očekivati da će korisnici imati nekakav pisač u pokretu, a istovremeno sve podatke vide dovoljno jasno na ekranu.

Moguća su daljnja proširenja aplikacije sukladno idejama i potrebama suvremenog poslovanja.

Rad je napravljen primjenom znanja stečenih kroz studij kao i znanja stečenih kroz dugogodišnje iskustvo u praksi.

LITERATURA

[1] Patrick LeBlanc – Microsoft SQL Server 2012 Step by step, Microsoft Press, 2013.

[2] Matt Doyle – Beginning PHP 5.3, Wiley Publishing, Inc., 2009.

[3] Damian Wielgosik - How to code in HTML5 and CSS3, free online book, 2015.

[4] https://hr.wikipedia.org [pristupljeno 20.06.2016.].

[5] http://www.fer.hr [pristupljeno 20.06.2016.].

SAŽETAK

Razvojem komunikacijskih tehnologija i povećanjem potrebe za bržim donošenjem poslovnih odluka razvila se potreba izrade aplikacije koja će u realnom vremenu omogućiti udaljeni pristup knjigovodstvenim podacima i na taj način postaviti osnovu za donošenje poslovnih odluka.

Kako sva poduzeća već imaju nekakve informatičke sustave i baze podataka koje se svakodnevno ažuriraju ideja je bila iskoristiti postojeću bazu podataka i korištenjem web tehnologija kreirati jednostavna i lako dostupna izvješća bilo gdje da se korisnik nalazi. U tu svrhu u izradi ovog rada korištene su web tehnologije otvorenog koda poput HTML5, CSS, PHP, Javascript i jQuery. Aplikacija je izrađena na način da bude prilagodljiva prijenosnim uređajima poput tableta i pametnih telefona, jer su to uređaji koji menadžeri najčešće imaju pri ruci.

Zaključno, izrađena aplikacija predstavlja potpunu implementaciju ideje i služi svojoj namjeni u potpunosti.

Ključne riječi: web aplikacija, poslovanje, HTML, CSS, PHP, Javascript, jQuery

ABSTRACT

WEB APPLICATION - COMPANY MONITORING BY THE MANAGER

The development of communication technology and the increasing need for faster business decision-making has developed the need to develop an application that will allow real-time remote access to accounting data and thus lay the basis for making business decisions.

As all companies already have some IT systems and databases that are updated daily, idea was to use the existing database and the use of web technology to create a simple and easily accessible reports anywhere that the user is located. For this purpose, in the preparation of this work were used Web technologies open source, such as HTML5, CSS, PHP, Javascript and jQuery. The application was made in a way that is flexible portable devices such as tablets and smartphones, as these are devices that managers usually have on hand.

In conclusion, made application represents the full implementation of ideas and serves its purpose fully.

Keywords: web application, management, HTML, CSS, PHP, Javascript, jQuery

ŽIVOTOPIS

Igor Balaš je rođen 7. svibnja 1971. godine u Vinkovcima gdje je završio osnovnu školu i Gimnaziju matematičkog smjera. Godine 1990. upisuje Elektrotehnički fakultet u Zagrebu i odlazi na odsluženje vojnog roka. Nakon povratka pokušava studirati na upisanom fakultetu, ali mu to ne uspijeva zbog poratnih i financijskih okolnosti. Tokom 1994. godine zapošljava se u informatičkom sektoru u jednoj privatnoj tvrtki kao programer u kojoj radi do kraja 1997. godine. Tada prelazi u Vinkovački vodovod na mjesto voditelja informatičke službe, a 1. kolovoza 1999. otvara obrt pod nazivom B computers. Od 1.veljače 2000. godine radi isključivo privatni posao. Od početka rada u struci bavi se programiranjem baza podataka i izradom knjigovodstvenih programa. Do 2004. programira koristeći CLIPPER, a tada prelazi na Microsoft SQL bazu podataka i Borlandov razvojni alat Delphi. I danas koristi navedene tehnologije samo u novijim inačicama. Posjeduje potpuna knjigovodstvena rješenja za poduzeća i obrte, veleprodaju, maloprodaju te male specijalizirane radnje. Aktivno se koristi engleskim i njemačkim jezikom. Od 9. godine života igra tenis u kojem se dugi niz godina i natjecao. Svira gitaru i klavir.