Razvoj PHP aplikacije uz podršku Oracle baze podataka

Pandi, Filip

Undergraduate thesis / Završni rad

2017

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:200:970398

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij / Stručni studij

RAZVOJ PHP APLIKACIJE UZ PODRŠKU ORACLE BAZEPODATAKA

Završni rad

FILIP PANDI

Mentor: Doc. dr. sc. Damir Filko

Osijek, 2017 godina.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Osijek, 21.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Ime i prezime studenta:	Filip Pandi
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI 4184, 25.10.2014.
OIB studenta:	33331502118
Mentor:	Doc.dr.sc. Damir Filko
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Emmanuel-Karlo Nyarko
Član Povjerenstva:	Petra Đurović
Naslov završnog rada:	Razvoj PHP aplikacije uz podršku Oracle baze podataka
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak završnog rada	Usporediti MySQL i Oracle sustave upravljanja bazama podataka te razviti PHP aplikaciju koja koristi Oracle bazu podataka.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	21.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 27.09.2017.

Ime i prezime studenta:	Filip Pandi
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI 4184, 25.10.2014.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Razvoj PHP aplikacije uz podršku Oracle baze podataka**

izrađen pod vodstvom mentora Doc.dr.sc. Damir Filko

i sumentora

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

Potpis studenta:

SADRŽAJ

1.	**UVOD**	1
	1.1 Zadatak rada	1
2.	**KORIŠTENE TEHNOLOGIJE U APLIKACIJI**	2
	2.1. Xampp	2
	2.1.1. Povezivanje Oracle baze podataka i Xampp PHP	2
	2.2. Oracle baze podataka	3
	2.2.1. PL/SQL	3
	2.2.2. Usporedba MySQL i Oracle PL/SQL	5
	2.2.3. Oracle SQL Developer	6
	2.3. Visual studio Code	6
	2.4. HTML	7
	2.5. CSS	7
	2.6. JavaScript	7
	2.7. PHP	7
3.	**STRUKTURA KORIŠTENE BAZE PODATAKA**	8
4.	**STRUKTURA RAZVIJENE APLIKACIJE**	11
	4.1. Registracija i autorizacija korisnika	11
	4.2. Unos i ispis rezultata	14
5.	**TESTIRANJE**	17
6.	**ZAKLJUČAK**	18
	LITERATURA	19
	SAŽETAK	20
	ABSTRACT	20
	ŽIVOTOPIS	21

1. UVOD

Tema završnog rada je PHP Web aplikacija uz Oracle bazu podataka na strani poslužitelja. Cilj rada je naučiti i praktično koristiti znanja o radu na strani poslužitelja zajedno sa bazom podataka. Zadatak se sastoji od dva dijela, dio na strani korisnika i dio na strani poslužitelja. Fokus je stavljen na poslužiteljsku stranu zadatka, a zajedno te dvije cjeline čine kompletnu aplikacijsku formu. U nastavku biti će opisana aplikacija, programski jezici i podrške koje su korištene, sama funkcionalnost aplikacije, dijelovi aplikacije te sam cilj.

Korišteni alati i razvojna okruženja prilikom izrade aplikacije su Xampp, Microsoft Visual Studio Code i SQL Developer, a korisnički dio i baza podataka su realizirani pomoću jezika HTML, CSS, JavaScript, PHP i PL/SQL.

Konačni cilj aplikacije je omogućiti krajnjem korisniku registraciju, prijavu te obračun i potraživanje kredita u veoma jednostavnom obliku. Ciljana skupina korisnika bile bi financijske ustanove koje imaju određenu bazu korisnika kojima je potrebna usluga kreditiranja.

Struktura rada podijeljena je u šest poglavlja. Uvodno poglavlje opisuje sam rad kao i njegovu problematiku. Drugo poglavlje opisuje korištene alate, sučelja i programske jezike potrebne za izradu aplikacije. Treće poglavlje opisuje samu strukturu baze podataka i korištene algoritme za izradu. Četvrto poglavlje govori o programskoj implementacijii načinima rješavanja problema u korisničkom dijelu aplikacije. Peto poglavlje sadrži i opisuje rezultate testiranja. Šesto poglavlje obuhvaća zaključak o cijelom projektnom zadatku.

1.1. Zadatak rada

Povezati Oracle bazu podataka na strani poslužitelja sa sučeljem namijenjenom korisniku pomoću PHP programskog jezika. Stvoriti aplikaciju koja omogućuje korisniku registraciju, prijavu, obradu unešenih podataka te funkcionalnost pregleda unosa prema različitim kriterijima. Naposljetku testirati rad cjelokupnogsustava.

2. KORIŠTENE TEHNOLOGIJE U APLIKACIJI

2.1.Xampp

Xampp je višeplatformski mrežni server koji omogućuje instalaciju i konfiguraciju različitih sustava mrežnih servera, a opisane su samim njegovim imenom*Cross-Platform, Apache, MySQL, PHP i Perl*.Jednostavan Apache server koji Xampp sadrži glavni je alat za stvaranje lokalnog*http* servera, odnosno lokalnog web servera koji koristiLAN mrežu određenog računala na kojemu je instaliran.Povezivanje se vrširučno uz mogućnost odabiranja proizvoljnog priključkaračunala. Također podržava Oracle proširenje u sklopu PHP podrške koja se naziva *OCI*. Osim navedenih primarnih funkcijaomogućava MySQL, FileZilla, Mercury, MariaDB i Tomcat podršku. Navedene platforme nije potrebno detaljnije opisivati s obzirom da nisu bitne za ovaj projekt.

2.1.1. Povezivanje Oracle baze podataka i Xampp PHP

Xampp PHP podržava različite sustave za upravljanje bazama podataka, a među njima je i Oracle baza podataka. Potrebno je instalirati Oracle bazu podataka na računalo te je prilagoditi potrebama. Nadalje potrebno je omogućiti *ocidll* pakete unutar Xampp PHP *ext* direktorija. *Oci* datoteke je potrebno postaviti u *ext* direktoriji (primjerice *C:\xampp\php\ext*).
Pomoću PHP*config*datoteke (*php.ini*) osposobiti slijedeće nadopune*extension=php_oci8.dll, extension=php_oci8_11g.dll, extension=php_oci8_12c.dll*. Zavisno o Oracle bazi podataka koja je instalirana.Nadopune se mogu osposobiti tako da se ukloni*";"* ispred nadopune. Naposljetku odrađuje se provjera da li su nadopune uspješno instalirane tako da se u tekstualnom editoru otvori PHP dokument koji sadrži liniju koda "*echo phpinfo();"*. Tako se dobiva potvrdna informacija da li je Oracle baza podataka uspješno povezana sa PHP-om. Na*phpinfo* stranici mora postojati *oci8* podrška kako bi Oracle baza podataka i PHP bili povezani.

2.2. Oracle baze podataka

Oracle baza podatakaje sustav za upravljanje bazama podataka (*eng. Relational Databse Managment System ili RDBMS*). Omogućuje konkretno sakupljanje i obradu podataka od strane klijenata ili korisnika nekakve usluge ili aplikacije. Podržan je od strane većineoperacijskih sustava (Linux, MS Windows, Unix). Koristi PL/SQL strukturalni jezik za upite prema bazama podataka uz proceduralno proširenje.

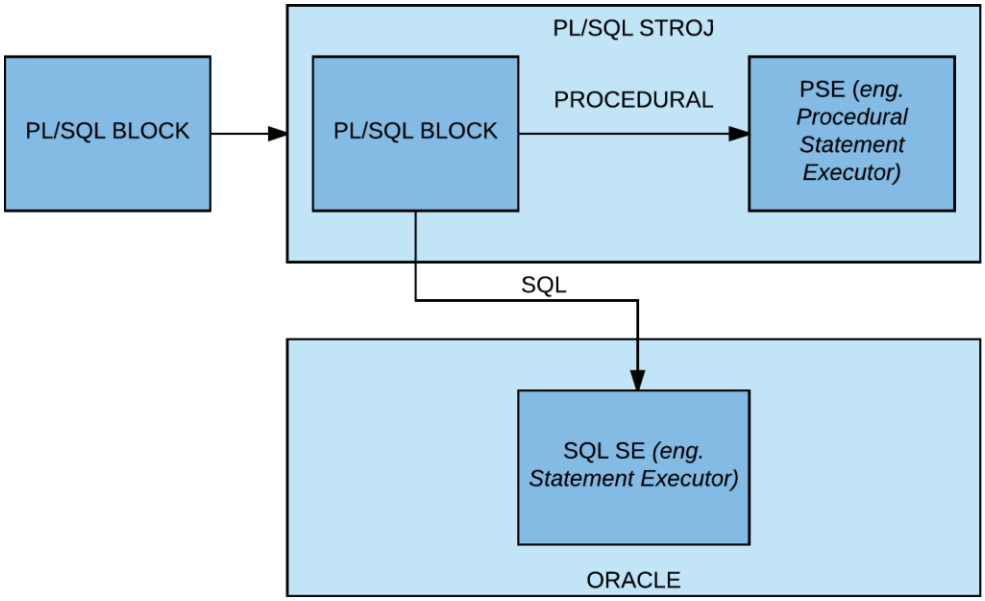
Oracle baza podataka kao i bilo koja druga baza podataka sastoji se odtablica, entiteta i veza. Entitet je osnovni element u bazi podataka predstavljen podatkom iz stvarnog svijeta koji želimo pohraniti.Tablice i entiteti mogu se povezivati različitim vezama, dijele se na relacije jedan naprema više,jedan naprama jedan i više naprema više.

2.2.1. PL/SQL

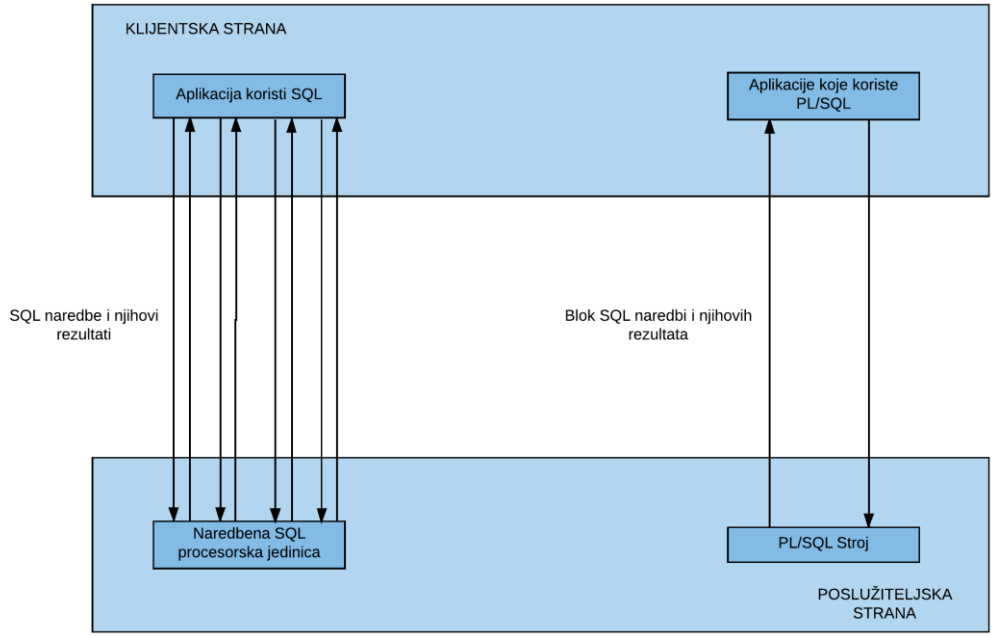
PL/SQL (*eng. Procedural Language/Structured Query Language*) (Slika 2.1.) je strukturalni jezik za upite prema bazi podataka uz proceduralno proširenje i samo stvaranje relacija unutar baze podataka. PL je proceduralno proširenje SQL jezika u Oracle bazama podataka. Omogućuje korištenje svih SQL naredbi kao što su funkcije, procedure, okidači i podržani tipovi podataka. Proceduralne osobine su značajkeprogramskih jezika (petlje, deklaracija varijabli, strukture i slično).

Naredbe koje se izvršavaju putuju od klijenta do baze podataka na poslužitelju, pa nakon izvršavanja vraćaju krajnji rezultat korisniku. Svaka SQL naredba putuje od klijenta do baze podataka također povratna informacija isto tako putuje od poslužitelja do korisnika. Bitna i značajna prednost je da ako postoji više naredbi u jednom PL/SQL bloku šalju se kao jedna cjelina na poslužitelj i vraćju se natrag korisniku, u obliku rezultata, isto tako kao cjelina te se na taj način smanjuje potrebno vrijeme izvršavanja naredbi. (Slika 2.2.)

Razlika između osnovnog SQL-a i PL/SQL-a je način na koji se izvode. Tako da je za izvođenje SQL naredbi zadužen SQL SE (*eng. Statment Executor*), a za izvođenje PL/SQL bloka od više naredbi PL/SQL stroj (*eng. PL/SQL Engine*). U jeziku PL/SQL svaka deklaracija, izvršavanje više naredbi, procedura, funkcija, potprograma, iznimaka obrađuje sekao blok naredbi te se tako šalje i prima.Na taj način štedi vrijeme potrebno za obradu kod kompleksnih baza podataka.



Slika 2.1. Obrada PL/SQL Bloka



Slika 2.2. Komunikacija između poslužiteljske i klijentske strane (SQL,PL/SQL)

2.2.2. Usporedba MySQL i Oracle PL/SQL

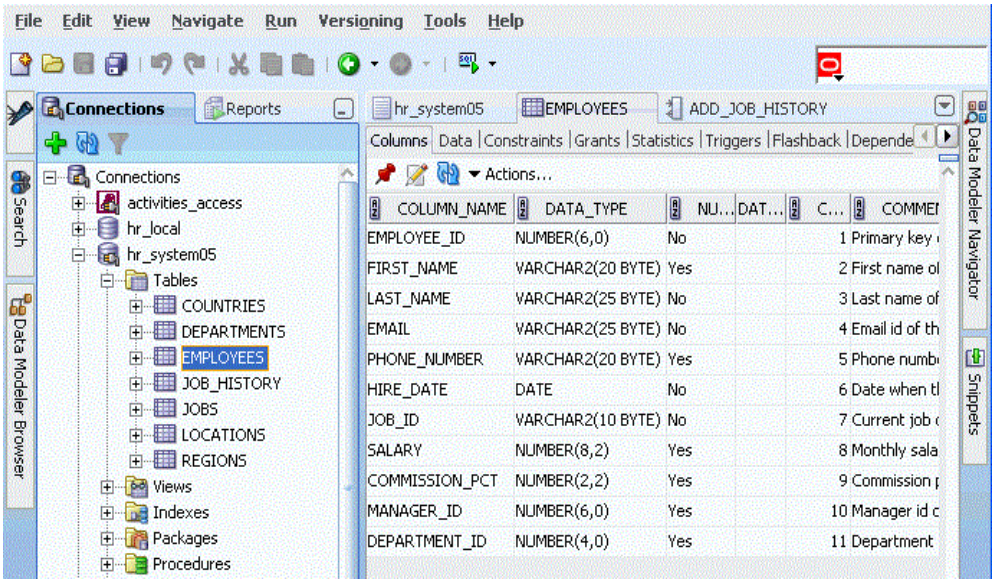
Kao što je već navedeno u prethodnom pod poglavlju, glavna razlika je što PL/SQL posjeduje proceduralno proširenje za deklaraciju i izvršavanje proceduralnih naredbi kao što su operatori, tipovi podataka i slično. Ako postoji više naredbi one se na poslužitelj šalju u obliku bloka jednom vezom jednako tako se i vraćaju korisniku. Također ima veliki broj predefiniranih SQL paket i visoku razinu sigurnosti. Važno je naglasiti da su oba jezika SQL jezici i da ne ovise o platformi na kojoj su instalirani.Sintakse ova dva jezika jako malo se razlikuju, gotovo ništa, zato što su zasnovane na ADA i Pascal programskim jezicima. Ako se nauči jedan jezik nikakav problem nije raditi u drugome i obrnuto.

Tablica 2.1. Usporedba Oracle i MySQL sistema

Kratka usporedba	
ORACLE	**MySQL**
Moćan alat za izradu baze podataka	Dobar alat za male projekte
Za velike OLTP (*eng. Online transaction processing*), VLDB (*eng.Very Large Database*)baze podataka	Za Web razvoj, male i srednje projekte
Nije besplatan	Besplatan
Puno opcija za korištenje	Limitirane opcije za korištenje
Brzina izvršavanja velika	Brzina izvršavanja prihvatljiva
Potrebno puno znanja da bi se iskoristio sav potencijal alata	Jednostavno rukovanje i instalacija

2.2.3. Oracle SQL Developer

Oracle SQL Developer (Slika 2.4.) je vrsta integriranog razvojnog okruženja koja omogućava razvoj, održavanje i upravljanje bazama podatka na poslužiteljskoj strani. Razvijen je od strane tvrtke Oracle isključivo za Oracle baze podataka 10g, 11g i 12c. Podržava programske jezike SQL i PL/SQL te pruža podršku za neke druge jezike kao što su Java i XML. Sadržava već predodređenepotprogramske pakete koji olakšavaju rad. Veoma je jednostavan i poprilično efikasan u izvršavanju većine operacija koje nudi.



Slika 2.3. Razvojno sučelje SQL Developer

2.3. Visual Studio Code

Visual Studio Code je editor izvornog koda razvijen od Microsoft kompanije za Windows, Linux i Mac operacijske sustave. Omogućava brzo i efikasno pronalaženje pogrešaka, izdvajanje ključnih riječi sintakse, nadopunjavanje koda prilikom pisanja i mnogo više. Podržava gotovo sve trenutno popularne programske jezike i razvojne cjeline (*eng. framework*) za razvoj aplikacija kao što su Python, Angular, #C, Java, HTML, CSS, PHP itd.

2.4. HTML

HTML (*eng. HyperText Markup Language*) je standardizirani jezik za razvoj korisničkog dijela web stranice (*eng. frontend*). Definira komponente web stranice kao što su zaglavlje, tijelo i podnožje te što će se nalaziti u njima. Podržava različite razvojne cjeline i biblioteke.

2.5. CSS

CSS (*eng. Cascading Style Sheets*) je stilski jezik koji služi za razvoj web stranicena strani korisnika. Definira samu vizualnu strukturu web stranice kao što su boje, umetanje slika, pozicioniranje objekata i drugo. Najpoznatija razvojna cjelina za CSS je Bootstrap.

2.6. JavaScript

JavaScript je vrsta skriptnog jezika koja omogućava vršenje raznih matematičkih i logičkih operacija te dohvaćanja vrijednosti varijabli i slično. Stvorenje po uzoru na Java objektno orijentirani programski jezik, radi lakšeg korištenja. No sam JavaScript nije objektno orijentirani jezik. Koristi samo jednu nit (*eng. Thread*) za komunikaciju sa poslužiteljem. AJAX (*eng. Asynchronous JavaScript and XML*) je razvojna cjelina koja omogućuje web stranicama komunikaciju sa poslužiteljskim programom uz podršku XML jezika za označavanje podataka.

2.7. PHP

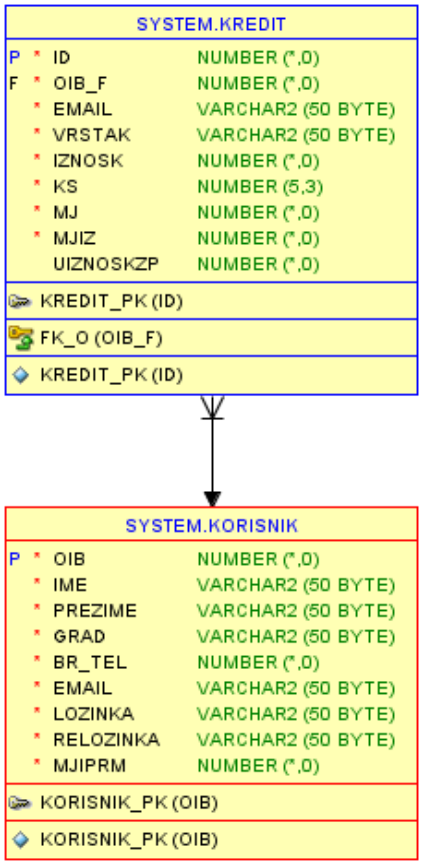
PHP (*eng. Hypertext Preprocessor*) je proceduralni, objektno orijentirani, skriptni programski jezik razvijen na temelju sintakseobjektno orijentiranog programskoj jezika Perl i sintaksi proceduralnog programskog jezika C. Koristi se za razvoj poslužiteljskog dijela, odnosno za povezivanje, obradu i pohranu podataka na poslužiteljskom dijelu u bazi podataka. Podržava razne sustave baza podataka kao što su Oracle bazapodataka, MySQL i druge. Može biti implementiran unutar HTML dokumenta ili dohvaćati vrijednosti pojedinih elemenata HTML dokumenta pomoću metoda "*post*" i "*get*".

3. STRUKTURA KORIŠTENE BAZE PODATAKA

Baza podataka sastoji se od dvije tablice, a to su tablica*KORISNIK* i tablica*KREDIT*. Povezane su vezom jedan naprama više (jedan korisnik može zatražiti više kredita ili više kredita može biti zatraženo od jednog korisnika). Veza između relacija ovako je orijentirana iz razloga što se koriste PHP sesije za pojedinog korisnika pa neće doći do pogrešaka u bazi prilikom unosa. Relacija *KORISNIK* predstavlja registrirane korisnike, a relacija *KREDIT* predstavlja odabrani kredit od strane korisnika.

Stvoren je pogled (*eng. View*) na relacije *KREDIT* i *KORISNIK*. Pogled povezuje ukupan iznos za otplatu i mjesečni dohodak korisnika koji se na kraju uspoređuju na stranici. (Slika 3.1.)

Baza podataka je stvorena koristeći Oracle SQL Developer *IDE* (*eng. Integrated Development Environment*).



Slika 3.1. E-R Dijagram baze podataka stvoren u alatu SQL Developer

Relacija *KORISNIK* sastoji se od devet atributa:

- *OIB* - Osobni identifikacijski broj korisnika, primarni ključ relacije
- *IME* - Ime korisnika
- *PREZIME* - Prezime korisnika
- *GRAD* - Grad u kojemu korisnik živi
- *BR_TEL* - Broj telefona korisnika
- *EMAIL* - Elektronička adresa korisnika
- *LOZINKA* - Lozinka za prijavu korisnika
- *RELOZINKA* - Ponavljanje lozinke zbog sigurnosnih razlog
- *MJIPRM* - Mjesečni prihodi korisnika

Relacija *KREDIT* sastoji se od devet atributa:

- *ID* - Primarni ključ relacije (AUTOINCREMENT)
- *OIB_F* - Strani ključ referiran na relaciju *KORISNIK*
- *EMAIL* - Email korisnika (Sesija prijave pomoću PHP-a)
- *VRSTAK* - Vrsta kredita
- *IZNOSK* - Iznos kredita
- *KS* - Kamatna stopa
- *MJ* - Mjesečni period otplate
- *MJIZ* - Mjesečni iznos otplate
- *UIZNOSKZP* - Ukupan iznos kredita za otplatu

Slika 3.2. i Slika 3.3.prikazuje SQL izvorni kod zastvaranje navedenih tablica. Koristi se uobičajena sintaksa za stvaranje baze podataka u SQL jeziku. Razvojno sučelje Oracle SQL Developer samo predlaže rješenja i prilagođava podatak prilikom stvaranja.

```
CREATE TABLE "SYSTEM"."KORISNIK"
        ("OIB" NUMBER(*,0) NOT NULL ENABLE,
         "IME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "PREZIME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "GRAD" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "BR_TEL" NUMBER(*,0) NOT NULL ENABLE,
         "EMAIL" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "LOZINKA" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "RELOZINKA" VARCHAR2(50 BYTE) NOT NULL ENABLE,
         "MJIPRM" NUMBER(*,0) NOT NULL ENABLE,
          CONSTRAINT "KORISNIK_PK" PRIMARY KEY ("OIB"); )
```

Slika 3.2. Izvorni kod tablice*KORISNIK* SQL Developer alata

```
CREATE TABLE "SYSTEM"."KREDIT"
        ("ID" NUMBER(*,0) NOT NULL ENABLE,
        "OIB_F" NUMBER(*,0) NOT NULL ENABLE,
        "EMAIL" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "VRSTAK" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "IZNOSK" NUMBER(*,0) NOT NULL ENABLE,
        "KS" NUMBER(5,3) NOT NULL ENABLE,
        "MJ" NUMBER(*,0) NOT NULL ENABLE,
        "MJIZ" NUMBER(*,0) NOT NULL ENABLE,
        "UIZNOSKZP" NUMBER(*,0) DEFAULT NULL,
         CONSTRAINT "KREDIT_PK" PRIMARY KEY ("ID");
```

Slika 3.3. Izvorni automatizirani kod relacije *KREDIT* SQL Developer alata

```
CREATE SEQUENCE  "SYSTEM"."ES"  MINVALUE 1 MAXVALUE 10000
INCREMENT BY 1;
```

Slika 3.4. Sekvenca *ES*: automatsko uvećanje atributa *ID*

Slika 3.4. prikazuje stvaranje sekvence za uvećavanje atributa *ID* relacije *KREDIT*.Uvećavanje je automatsko podizanje vrijednosti određenog atributa prilikom novog unosa u relaciju. Korak povećanja vrijednosti možemo proizvoljno odrediti, a možemo ograničiti sekvencu na minimalnu i maksimalnu vrijednost. Počinje zadanom vrijednošću te kada dostigne zadanu maksimalnu vrijednost dolazi do prelijevanja podataka.

```
CREATE OR REPLACE FORCE VIEW "SYSTEM"."KOKR" ("MJIPRM",
"UIZNOSKZP", "EMAIL") AS   SELECT KORISNIK.MJIPRM,
KREDIT.UIZNOSKZP, KORISNIK.EMAILFROM KORISNIK,KREDIT
WHERE KORISNIK.EMAIL=KREDIT.EMAIL;
```

Slika 3.5. Pogled *KOKR* koji spaja tablice*KORISNIK* i *KREDIT*

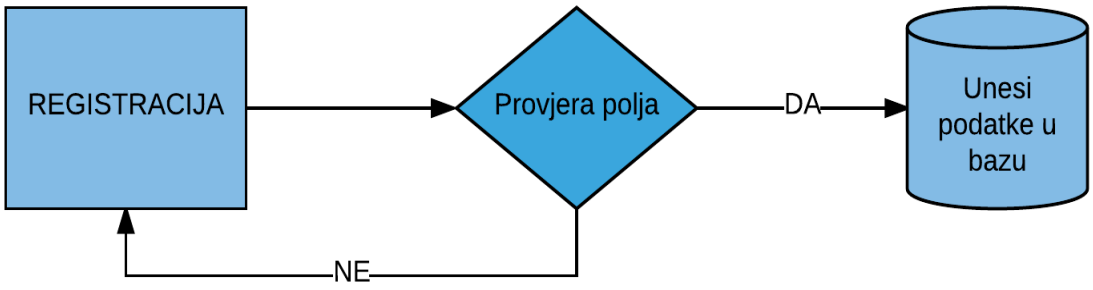
Slika 3.5. prikazuje stvaranje pogleda *KOKR*. Pogled uzima vrijednosti mjesečnih primanja, ukupnog iznosa kredita za otplatu i e-mail adrese iz relacija *KORISNIK* i *KREDIT* gdje je atribut *EMAIL* jedne tablicejednak *EMAIL*druge. PHP sesija (*eng. session*) dohvaća e-mail adresu prilikom prijave te ga sprema u lokalni spremnik sesije iz kojeg se kasnije dohvaća i pohranjuje u relaciji *KREDIT*. Jednako tako moguće je stvoriti sesiju pomoću XML programskog jezika na sličan način, jedina razlika je što XML jezik sprema kratkotrajne varijable unutar virtualnogspremnika *JSON*.

4. STRUKTURA RAZVIJENE APLIKACIJE

4.1. Registracija i autorizacija korisnika

Prvenstveno korisnik se mora registrirati kako bi koristio aplikaciju. Registracija je opisana pomoću obrasca koji sadrži polja za unos osnovnih informacija koje se pohranjuju u bazu podataka nakon pritiska na gumb. Korisniku se prezentira obrazac za prijavu gdje se prijavljuje sa podatcima koje je unio prilikom registracije.

Nakon što se korisnik prijavio prikazuje se stranica koja sadrži jednostavni oblik kamatnog kalkulatora kako bi korisnik mogao provjeriti koliki je mjesečni izdatak za pojedini iznos glavnice i kamatne stope.
Klikom na gumb, otvara se prozor koji je sakriven dok se ne pritisne gumb, koji sadrži polja za unos i potraživanje kredita. Klikom se podatci spremaju u bazu te su dostupni na korisničkom panelu.



Slika 4.1. Dijagram toka provjere podataka prilikom registracije

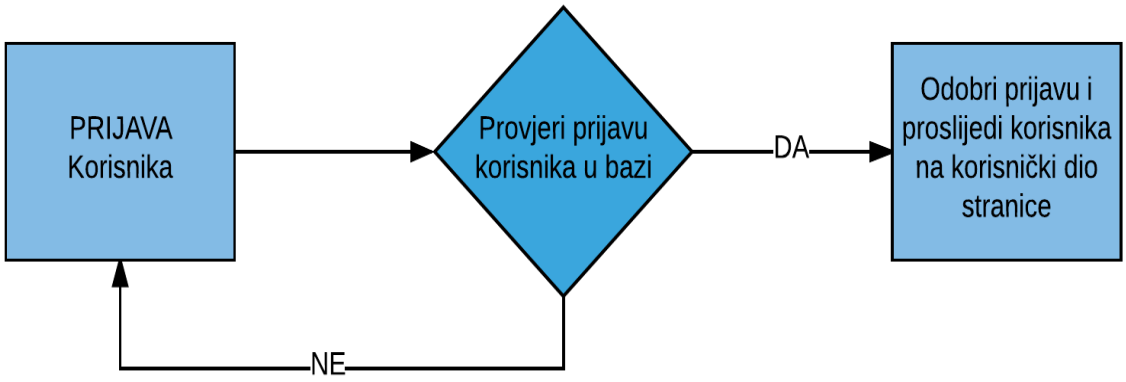
Slika 4.1. prikazuje dijagram toka registracijskog obrasca. PHP-om se provjerava da li su sva polja ispravno unešena, da li se određena polja podudaraju i da li postoje isti unosi u bazi podataka. Ako se dogodi neka pogreška ta pogreška biti će ispisana na ekran te će se korisnik preusmjeriti na registracijski obrazac.

```
... Dohvaćanje vrijednosti podataka POST metodom
if(!filter_var($email, FILTER_VALIDATE_EMAIL))
$error_msg[]='Email adresa nije valjana oblika!';
$select = "SELECT EMAIL FROM KORISNIK WHERE EMAIL = '$email'";
$result = oci_parse($connect, $select);
oci_execute($result);
$tmpcount = oci_fetch($result);
if($tmpcount){
$error_msg[]='Unešena email adresa već se koristi!';
echo 'Unešena email adresa već se koristi! <br>';}
$s = "SELECT OIB FROM KORISNIK WHERE OIB = '$oib'";
$result2 = oci_parse($connect, $s);
oci_execute($result2);
$tmpcount2 = oci_fetch($result2);
if($tmpcount2){
$error_msg[]='Korisnik s ovim OIB brojem već postoji!';
echo 'Korisnik s ovim OIB brojem već postoji!';}
if(count($error_msg) == 0){
$insert = "INSERT INTO KORISNIK
VALUES('$oib','$ime','$prezime','$grad','$telefon','$email','$lozinka','$relozinka','$osp')";
$r=oci_parse($connect, $insert);
$a=oci_execute($r);
.... Izvršavanje i upis u bazu, ako podatci nisu valjani povratak na registracijski obrazac
```

Slika 4.2. Izvorni kod provjere i unosa podataka

Slika 4.2. prikazuje kod za provjeru unesenih podataka u polje za unos na stranici registracija. Vrijednosti polja za unos dohvaćaju se metodom "*post*" te se vrijednosti šalju na obradu. Uz pomoć spremnika polja*"error_msg[]"*, svaka se pogreška pri unosu bilježi u to polje. *if* naredba provjerava veličinu polja, odnosno da li su se dogodile pogreške i da li su zapisane u polje. Ako je broj pogrešaka nula podatci se upisuju, a ako pogreška postoji ispiše se pogreška koja je nastala.

Od korisnika se traži elektronička adresa i lozinka kako bi se izvršila prijava. Vrijednosti elektroničke adrese i lozinke uspoređuju se sa postojećim podatcima u bazi.Ako su podatci valjani korisnik se prosljeđuje dalje na korisnički dio stranice. Pomoću php *session* ugrađene funkcije dohvaćaju se uneseni podaci te se pamte dok god se korisnik ne odjavi sa svog korisničkog profila.



Slika 4.3. Dijagram toka prijave korisnika

Slika 4.4. prikazuje blok naredbi koje omogućuju otvaranje nove sesije za pojedinog korisnika koji se prijavljuje. Uz dohvaćanje podataka iz baze te uspoređivanje podataka sa unesenim vrijednostima.

Slika 4.5.opisuje blok naredbi za odjavu korisnika koji se nakon odjave prosljeđuje na stranicu za prijavu.

```
...
session_start();
  if($_SERVER["REQUEST_METHOD"] == "POST") {
     $eml = $_POST['kor'];
     $lozi = $_POST['loz'];
     $sql = oci_parse( $connect, "SELECT * FROM KORISNIK WHERE EMAIL = '$eml'
and LOZINKA = '$lozi'");
     oci_execute($sql);
     if(oci_fetch_array($sql)) {
        $_SESSION['lu'] = $eml;
...
```

Slika 4.4. Dio izvornog koda za prijavu korisnika

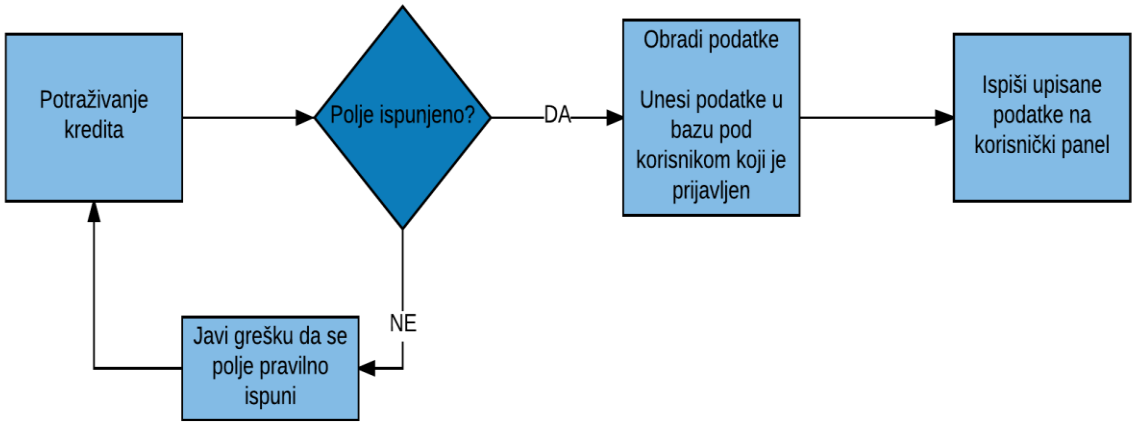
```
session_start();
session_destroy();
header('Location: index.php');
exit;
```

Slika 4.5. Izvorni kod odjava korisnika

4.2. Unos i ispis rezultata

Stranica koja se pojavljuje nakon prijave omogućuje korisniku provjeru mjesečnog izdatka otplate pomoću jednostavnog kamatnog kalkulatora te potraživanje kredita koji se nakon obrade pohranjuje u bazu podataka.

Rezultati zatraženog kredita prikazuju se na korisničkom panelu koji ispisuje sve informacije vezane uz zatraženi kredit. Korisnik je u mogućnosti zatražiti kredit u iznosu dvije trećine mjesečnih prihoda. Korisnik može zatražiti onoliko kredita dok god suma mjesečnih prihoda ne dostigne dvije trećine mjesečnih prihoda korisnika. Kako bi kredit bio u potpunostiodređen potrebno je fizičko odobrenje.



Slika 4.6. Dijagram toka potraživanja kredita

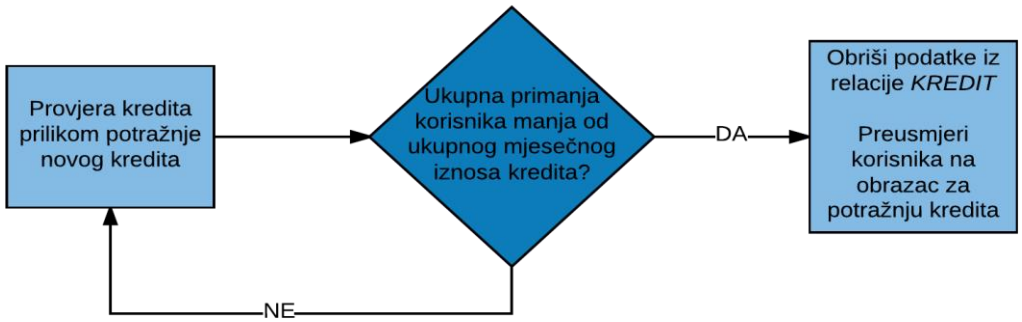
Obrazac za potraživanje kredita otvara se pritiskom na gumb, a sastoji se od tri polja. Trošni iznos kredita, vrsta kredita i vrijeme otplate u mjesecima. Polje vrijeme otplate ne može se ručno unositi već se popunjava na temelju unešenog iznosa kredita. Polje vrsta kredita je padajući izbornik koji sadrži različite vrste kredita te odabirom na pojedinu vrstu kredita dodaje se druga kamatna stopa na iznos kredita.

Slika 4.7.prikazuje dio izvornog koda koji radi pomoću *switch* naredbe, teizdvaja odabranu vrstu kredita iz padajućeg izbornika. Računa iznos kredita i glavnicu sukladno o odabiru vrste kredita te podatke upisuje u bazu podataka.

```
...
if(isset($rad)){
    switch($rad){
    case 'a':
    $mjiz= ($ts * 0.0887)/$pa;
   $uk = (($ts / $pa) + $mjiz);
 $tp = $uk *$pa;
 $sqli = oci_parse($connect, "SELECT OIB FROM KORISNIK WHERE EMAIL = '".$_SESSION['lu']."'");
      oci_execute($sqli);
$sw=oci_fetch_assoc($sqli)['OIB'];
     $insert = "INSERT into KREDIT ( ID,OIB_F, EMAIL, VRSTAK, IZNOSK, KS, MJ, MJIZ,UIZNOSKZP)VALUES
(ES.NEXTVAL,'$sw' ,'".$_SESSION['lu']."','Gotovinski','$ts','8.87','$pa','$uk','$tp')";
$r=oci_parse($connect, $insert);
$a=oci_execute($r);
...
```

Slika 4.7. Izvorni kod za unos i obradu kredita

Nadalje korisnik se prosljeđuje na korisnički panel gdje su prikazani svi zatraženi krediti i rezultati obrade. Ako mjesečna potraživanja kredita premaše dvije trećine mjesečnih primanja korisnika te ako i ukupna mjesečna potraživanja premaše dvije trećine mjesečnih primanja korisnika, prikazuje se poruka igumb za ponovno potraživanje kredita. Gumb za ponovno potraživanje kredita briše podatke iz baza podataka te preusmjeravakorisnika na obrazac za potraživanje kredita.



Slika 4.8. Dijagram toka provjere kredi

```
...
$vie2= oci_parse ($connect, "SELECT SUM(MJIZ) AS MIS, MJIPRM AS MJIE FROM KOKR WHERE EMAIL = '".$_SESSION['lu']."' GROUP BY MJIPRM");
oci_execute($vie2);
while($row5=oci_fetch_assoc($vie2)){
if($row5['MIS'] < $row5['MJIE'])
{
$sqli2 = oci_parse( $connect, "SELECT SUM(UIZNOSKZP) AS V, SUM(MJ) AS M, SUM(MJIZ) AS MI FROM KREDIT WHERE EMAIL = '".$_SESSION['lu']."'");
oci_execute($sqli2);
echo "<h3 id='hf1'><u>Vrijednosti za odplatu: </u></h3>";
echo "<table border=1 class='table table-striped' id='t'>";
echo "<tr><td><strong>Ukupni Iznos Kredita (Kn)</strong></td><td><strong>Ukupan Broj Mjeseci</strong></td><td><strong>Ukupan Mjesečni Izdatak</strong></td></tr>";
while($row2=oci_fetch_assoc($sqli2)){
echo "<tr><td>$row2[V]</td><td> $row2[M]</td><td> $row2[MI]</td></tr>";
}
}
else{
echo '<h1 id="k1"> Vaš kredit nije odobren, vaš ukupan mjesečni izdatak je veči od dozvoljenog iznosa.</h2>';
echo '<form name="por" method="post" action="Del.php"><button type="submit" class="btn btn-secondary btn-md" name="del" id="bet"><strong>Zatražite Ponovno!</strong></button></form>';
...
```

Slika 4.9. Izvorni kod provjere, odobravanjakredita i brisanja podataka

Slika 4.9. prikazuje dio izvornog koda koji provjerava jesu li dvije trećine mjesečnih primanja korisnika veća odukupnog mjesečnog iznosa kredita. Problem je jednostavno riješen pomoću *if* naredbe koja dohvaća sumu mjesečnog kredita i mjesečna primanja korisnika iz kreiranog pogleda *KOKR* koji spaja dvije relacije *KORISNIK* i *KREDIT* te ih uspoređuje. Ako je suma zatraženog kredita manja od dvije trećine ukupnih mjesečnih primanja korisnika ispisati će se vrijednosti u tablicu na stranici iz baze podataka. Ako je suma zatraženog kredita veća od dvije trećine ukupnih mjesečnih primanja korisnika obriši podatke u relaciji *KREDIT* te uputi korisnika da zatraži novi kredit na obrazac za potraživanje kredita.

5. TESTIRANJE

Testiranje je izvršeno na jednostavnoj razini. Broj korisnika koji su testirali aplikaciju je mali, četiri korisnika uspješno je koristilo aplikaciju za potraživanje kredita. Testiranje nije bilo potrebno vršiti na većem broju korisnika zbog jednostavnosti same aplikacije. Aplikacija omogućuje registraciju, prijavu i potraživanje kredita pojedinom korisniku.Očekivani rezultati su uspješno postignuti.

TEST ID	Akcija	Preduvjet	Rezultat
1.	Korisnik unosi svoje podatke na registracijskom obrascu	Korisnik mora imati otvorenu e-mail adresu i stalne mjesečne prihode	Korisnički podatci su upisani u bazu podataka
2.	Korisnik unosi e-mail adresu i zaporku na obrascu za prijavu	Korisnik ima aktivan korisnički račun	Korisnik se prijavljuje u sustav
3.	Korisnik unosi podatke u kamatni kalkulator (proizvoljno)	Korisnik ima aktivan korisnički račun	Korisnik dobiva informacije o mjesečnim izdatcima kredita i odlučuje se za potraživanje kredita
4.	Korisnik unosi iznos kredita u polje na obrascu za potraživanje kredita	Korisnik ima aktivan korisnički račun	Korisnik se prosljeđuje na korisnički panel
5.	Korisnik ima mogućnost pretraživati unoseći vrijednosti "*od*", "*do*" u tražilicu (proizvoljno)	Korisnik ima aktivan korisnički račun	Korisnik dobiva podatke koje je tražio
6.	Korisnik zatraži kredit veći od svojih mjesečnih primanja	Korisnik ima aktivan korisnički račun	Korisnik proslijeđen na obrazac za potražnju kredita
7.	Korisnik odabire odjavu	Korisnik ima aktivan korisnički račun	Korisnik proslijeđen na prijavu

6. ZAKLJUČAK

Oracle baza podataka i svi njeni alati su moćni i fleksibilni. Omogućuju stvaranje velikih i kompleksnih baza podataka. Brzina izvođenja uvelike je veća, zavisno o broju i kompleksnosti naredbi za izvršavanje, od drugih sličnih sustava za izradu i manipulaciju bazama podataka. Značajna mana je kompleksnost podešavanja i česte pogreškeprilikom instalacije i konfiguracije same baze podataka, nepravilan rad *TNS Listener-a*nakon ažuriranja Windows operativnog sustava i slični problemi. Također nije jeftina investicija. Za bilo kakav poslužiteljski rad potrebno je izdvojiti dosta novaca. Ciljana skupina korisnika Oracle baze podataka su velike tvrtke s velikim projektima, a zbog visoke cijene korištenja nije predviđena za privatnu upotrebui za izradu malih projekata.

Izrađena je aplikacija jednostavna. Potrabna je prilagodba za različite vrste bankovnih sustava. Različiti bankovni sustavi koriste različite vrste naknada i razne dodatne troškove koje je potrebno programski implementirati kako bi aplikacija zadovoljila potrebe pojedinih sustava.

U sklopu projekta napravljena je web aplikacija kojom se korisnik može registrirati, prijaviti u sustav, potraživati kredite sukladno postavljenim uvjetima korištenja i imati evidenciju svih zatraženih kredita.

Aplikacija je uspješno testirana na broju od četiri korisnika, zbog jednostavnosti same aplikacije, te radi u skladu s postavljenim očekivanjima.

LITERATURA

[1] Oracle Help Center, stranica: https://docs.oracle.com/en/
[2] Tutorialspoint: Oracle Tutorials, stranica: http://www.tutorialspoint.com/listtutorials/oracle/1
[3] PHP.net: OCI8 Functions, stranica: http://php.net/manual/en/ref.oci8.php
[4] StackOverflow: GIT forum, stranica: https://stackoverflow.com/
[5] SlideShare: Oracle MySQL comparison, stranica: https://www.slideshare.net/
[6] W3Schools: SQL, JavaScript, PHP, Bootstrap, stranica: http://www.w3schools.com

SAŽETAK

Ovaj rad opisuje razvoj aplikacije koja omogućuje registraciju korisnika, prijavu korisnika, potraživanje kredita, obradu i ispis podataka vezanih uz određenog korisnika.Cilj rada je praktično koristiti znanja o radu na strani korisnika i poslužiteljazajedno sa bazom podataka. Uz pomoć editora izvornog koda Visual Studio Code napisan je dio na strani korisnika u programskim jezicima PHP, HTML, CSS i JavaScript. Baza podataka na poslužiteljskoj strani stvorena je pomoću Oracle SQL Developer integriranog razvojnog okruženja i PL/SQL jezika.

Ključne riječi: Oracle, PHP, web aplikacija, kredit.

ABSTRACT

This application project allowes for user to register, login, request a loan andgetinforamtion about requested loans. Goal of this project is to developfrontend for users and backend with database structure. In Visual Studio Code frontend part of the project was created in programing languages PHP, HTML, CSS and JavaScript.Database on backend is followed out with integrated development environment Oracle SQL Developer and PL/SQL language.

Keywords: Oracle, PHP, web application, loan.

ŽIVOTOPIS

Filip Pandi rođen je 22.2.1993 godine u Osijeku, Republika Hrvatska. Pohađao je osnovnu školu Višnjevac od 2000 do 2008 godine. Upisao je III. Gimnaziju Osijek 2008 godine te je završio 2012 godine. Godine 2012 upisao je Elektrotehnički fakultet u Osijeku, smjer Informatika. Počeo je raditi u tvrtci Aspectus 2017godine na usavršavanju i preprodaji digitalnih usluga.