

RAZVOJ APLIKACIJE ZA EVIDENCIJU STUDENATA

SAŽETAK: Cjelokupan rad je baziran na razvoju C# aplikacije i njenim spajanjem sa bazom podataka, u ovom slučaju sa MySQL bazama. C# predstavlja programski jezik koji služi za izradu aplikacija koje mogu raditi pod .NET okruženjem. Za programiranje u C# korišten je Visual Studio .NET, koji je u mnogome olakšao razvoj same aplikacije. Visual Studio .NET pored toga što sadrži dizajnerske alate za aplikacije, alate za prikazivanje i navigaciju elemenata projekta, omogućava jednostavniju distribuciju i prilagođavanje projekta klijentima i obezbjeđuje napredne tehnike otklanjanja grešaka pri razvoju projekta. C# je potpuno objektno-orijentisani programski jezik, što znači da svi elementi unutar njega predstavljaju objekt. Kreirana je MySQL baza podataka u kojoj ćemo pronaći potrebne podatke, a kojoj će se moći pristupiti preko aplikacije. MySQL je besplatan, open source sustav za upravljanje bazom podataka. Ideja i cilj aplikacije bila je omogućiti učinkovitu i pouzdanu upravljačku bazu podataka za profesionalne potrebe.

KLJUČNE RIJEČI: C#, programski jezik, MySQL, baza podataka, Visual Studio

1. UVOD

U svijetu kakvog danas poznajemo sve se više i više susrećemo s tehnologijama koje nam život čine boljim tako da što god zamislimo ili nam predstavi problem, možemo olakšati korištenjem adekvatnog softvera ili alata. IT industrija od svojih začetaka bilježi kontinuirani rast i razvoj te je postala toliko napredna i uspješna da se danas aplikacije rade za stvari koje prije nismo mogli niti zamisliti. Kako je moguće razvijati aplikacije za različite platforme, za početak razvoja softvera potrebno je odgovoriti na pitanje što od alata i na koji način dobro iskoristiti? Kako u suvremeno doba skoro svaka obitelj posjeduje barem jedno računalo, odgovor na ovo pitanje rezultirao je odlukom o izradi računalne aplikacije (eng. „Desktop application”) u programskom okviru „.NET Framework“. Kako napreduje razvoj aplikacija u ovom području, normalno da je i konkurencija na tržištu velika stoga aplikacije s vremenom postaju sve bolje i iz njih se očekuje više, ali isto tako je i njihov razvoj sve više i više olakšan te prilagođen čovjeku što je postignuto kroz razne programske okvire.

Glavna inspiracija za ovu aplikaciju je popularnost programskog okvira koja daje veliki izbor mogućnosti razvoja aplikacija te samo učenje i napredak karijere u polju razvoja softvera. Osim „Microsoft Visual Studio“ i korespondirajućih alata koji dolaze uz ovo razvojno okruženje, za funkcionalan rad ove aplikacije zaslužan je i „MySQL“ na kojemu je smještena baza podataka. Nešto više o metodama i alatima opisano je u sljedećem poglavlju rada.

Aplikacija kao glavnu namjenu ima evidenciju podataka o studentima. Neke od glavnih funkcionalnosti aplikacije su da se korisnicima omogući spremanje bitnih podataka vezanih za studente i njihov napredak. Aplikacija sadrži i druge osnovne funkcionalnosti koje su neophodne za rad današnjih aplikacija kao što su registracija i autentikacija, izmjena i pregled vlastitih informacija te drugih stvari vezanih uz tematiku što je korisnicima ograničeno s određenim ulogama. Cjelokupna dokumentacija navedena je u praktičnom dijelu rada.

2. C# PROGRAMSKI JEZIK

C# je zreo programski jezik koji omogućava izradu aplikacija za Web sa svom potrebnom funkcionalnošću. Osim što pruža programski okvir za razvoj softvera, omogućava i podršku za pristupanje brojnim bazama podataka. On predstavlja jedan od jezika koji služe za izradu aplikacija koje mogu raditi pod .NET okruženjem. Kako predstavlja evoluciju jezika C i C++, kao takav je kreiran uz korištenje svih prednosti ostalih jezika i otklanjanje njihovih mana. Sam razvoj aplikacije je jednostavniji u C# jeziku, a sam jezik je izuzetno modan. S vremena na vrijeme njegov kod je mnogo razumljiviji u odnosu na druge jezike, ali je nešto robusniji i jednostavnije je otkloniti greške u njemu.

Sam jezik nema ograničenja u pogledu toga kakve se aplikacije mogu napraviti u njemu, on koristi okruženje i samim tim nema ograničenja u vezi sa mogućim aplikacijama. Za programiranje u C# korišten je Visual Studio .NET, koji je u mnogome olakšao razvoj same aplikacije. Visual Studio .NET pored toga što sadrži dizajnerske alate za aplikacije, alate za prikazivanje i navigaciju elemenata projekta, omogućava jednostavniju distribuciju i prilagođavanje projekta klijentima i obezbeđuje napredne tehnike otklanjanja grešaka pri razvoju projekta.

Programski jezik C# je proizvod kompanije Microsoft i nastao je kao odgovor na nedostatke postojećih jezika kao što su C, C++ i Visual Basic, u isto vrijeme kombinujući njihove dobre strane. C# je potpuno objektno-orientisani programski jezik, što znači da svi

elementi unutar njega predstavljaju objekt. Objekt predstavlja strukturu koja sadrži podatkovne elemente, kao i metode i njihove međusobne interakcije.

2.1. KLASI I OBJEKTI U C#

Do klasa dolazimo polazeći od pojedinačnih objekata. Posmatranjem objekata uočavamo njihova zajednička svojstva koja zovemo atributima. Takođe, objektima možemo pridružiti iste „akcije“ koje u okviru klase nazivamo metodima klase. Njima opisujemo funkcionalnosti objekata klase. Klasu definišemo navođenjem rezervisane riječi `class` iza koje slijedi identifikator klase (ime klase). Poslije imena, ako formiramo klasu koja nasljeđuje neku, prethodno definisanu klasu, navodimo `:` a zatim slijedi ime klase iz koje je izvedena. Zatim u vitičastim zagradama `{ }` definišemo članove klase (atributi i metodi).

Atributima opisujemo određenu osobinu objekta (ime, prezime, datumRodjenja, razred, odjeljenje). Najčešće, različiti objekti iste klase imaju različite vrijednosti atributa. Često kažemo da vrijednosti atributa definišu stanje objekta. Pri opisu atributa moramo navesti tip kome taj atribut pripada (celobrojni, realni, znakovni, Form, Button) i ime atributa. Pri tome, navedeni tip je prethodno definisan (ugrađen u sistem ili definisan od strane programera).

U programskom jeziku C#, po navođenju `/*`, pišemo komentare napisanog koda. Počev od `/*` pa do kraja linije napisani komentar se pri prevođenju programa ignoriše. Ukoliko pišemo opširniji komentar, koji obuhvata više redova, od ostalog dela koda odvajamo ga na početku sa `/*` a na kraju sa `*/`.

Jedan od osnovnih koncepata objektno-orijentisanog programiranja jeste kontrolisanje pristupa članovima objekta (atributima i metodima). Za svaki član klase (atribute i metode) moramo definisati nivo pristupa (vidljivosti, dostupnosti, zaštite). Na taj način definišemo koliko je taj član klase otvoren prema „spoljašnjem“ svetu. U programskom jeziku C# postoje više nivoa pristupa a mi ćemo, u početku, koristiti privatni (`private`) i javni (`public`) pristup. Privatnim članovima klase mogu pristupati samo metode i te klase i to je najviši nivo zaštite. Pri opisu javnih članova klase navodimo rezervisanu riječ `public`. Takvi članovi klase su dostupni svijetu van klase i pristup tim članovima je potpuno slobodan.

3. INTEGRIRANO RAZVOJNO OKRUŽENJE

Svaki programski jezik ima svoje integrirano razvojno okruženje koje omogućava lakše pisanje koda. Integrirano razvojno okruženje je softverska aplikacija koja pruža sveobuhvatne mogućnosti programerima za razvoj softverskih rješenja.

Razvojno okruženje se obično sastoji od uređivača izvornog koda, alata za automatizaciju gradnje i programa za ispravljanja grešaka. Mnoga moderna razvojna okruženja imaju i preglednik klasa, objekata i hijerarhijski dijagram klasa za upotrebu u objektno-orijentiranom razvoju programskih okruženja. Integrirana razvojna okruženja su osmišljena tako da maksimiziraju produktivnost programera pružajući čvrsto povezane komponente sa sličnim korisničkim sučeljima.

Rani sistemi nisu mogli podržati integrirana razvojna okruženja jer su se programi pripremali koristeći dijagrame toka. Prvi programski jezik koji je stvoren s integriranim razvojnim okruženjem je "Dartmouth BASIC". Njegovo razvojno okruženje se temeljilo na naredbama is toga nisu previše izgledali poput današnjih. "Maestro I" je proizvod tvrtke "Softlab Munich" i prvi program sa integriranim razvojem okruženjem koje je imalo grafičko korisničko sučelje, mogućnost uređivanja, upravljanja datotekama, ispravljanje grešaka u skladu sa modernim razvojnim okruženjima. Danas imamo razvojna okruženja poput "Microsoft Visual Studio", "NetBeans" i "Eclipse".

3.1. MICROSOFT VISUAL STUDIO

Nakon upoznavanja sa tim šta je u stvari jezik C# i kako se uklapa u .NET okruženje, došao je trenutak da započnemo sa pisanjem koda. U tom poslu koristimo Visual Studio .NET (VS), pa treba da naučimo osnovne stvari o tom razvojnom okruženju. VS je ogroman i komplikovan proizvod, tako da u prvo vrijeme korisniku može djelovati zastrašujuće, ali uz njegovu pomoć pravljenje jednostavnih aplikacija može biti iznenađujuće prosto. Čim počnemo sa korištenjem VS-a uvidate da nije potrebno mnogo znati o njemu da bismo se poigrali sa C# kodom. Kasnije ćemo videti neke mnogo komplikovanije operacije koje je moguće izvesti u VS-u, ali za sada su dovoljna osnovna znanja.

Microsoft Visual Studio je integrirano razvojno okruženje tvrtke Microsoft. Koristi se za razvoj računalnih programa, web usluga, web stranica, web aplikacija i mobilnih aplikacija. Visual Studio koristi platforme koje je razvila tvrtka Microsoft, a to su Microsoft Silverlight,

Windows Presentation Foundation, Windows Store, Windows API i Windows Form aplikacija. Uređivač koda, kojeg uključuje Visual Studio, podržava “IntelliSense” komponentu za dovršetak i refaktoriranje koda. Ostatak ugrađenih alata su dizajner za izgradnju grafičkog korisničkog sučelja, profiler koda, dizajn web usluga, dizajner baze podataka i dizajner klasa. Visual Studio prihvata dodatke na svakoj razini, odnosno mogu se uključiti dodaci za širok spektar platformi. Visual Studio razvojno okruženje podržava čak 36 programskih jezika. Najosnovnija verzija Visual Studio razvojnog okruženja je “Community” izdanje, koje je besplatno. Za razvoj ovog programa je korišten Microsoft Visual Studio Professional 2019.

3.2. .NET FRAMEWORK

.NET Framework je softverski okvir koji je razvio Microsoft, a koji prvenstveno radi na Microsoft Windows sistemu. To je najbitnija komponenta programa i programskog jezika koji smo koristili kako bismo preko Visual Studio programa povezali svoje baze na aplikaciju. Većinom sadrži uobičajene programske jezike koji nam pružaju lakše upravljanje memorijom i ostalim servisima našeg sistema.

NET Framework je revolucionarna platforma koju je kreirao Microsoft za razvoj aplikacija. To je takva platforma na kojoj mogu raditi različite vrste aplikacija (Windows aplikacije, Web aplikacije, Web servisi...). NET Framework je dizajniran tako da može koristiti bilo koji programski jezik kao što su C++, Visual Basic, Jscript, čak i ranije verzije Cobol. Takođe je moguće da se kombinuju pojedini dijelovi koda iz različitih programskih jezika. Npr. kod napisan u C# se može kombinovati sa kodom iz Visual Basica.

Uključuje veliku biblioteku koja se naziva “Framework Class Library” (FCL) i pruža mogućnost da se može koristiti na više programskih jezika. FLC pruža mrežne komunikacije, pristup podacima, povezivanje sa bazom podataka, korisničko sučelje i razvoj web aplikacija. “Common Language Runtime” (CLR) je softversko okruženje u kojem se izvršavaju program napisani za .NET Framework. To je zapravo virtualni stroj koji korisnicima pruža usluge rukovanja iznimkama, sigurnosti i upravljanja memorijom. Kada spojimo FCL i CLR dobijemo .NET Framework.

Postoje dvije vrste upotrebe .NET Framework-a, a to su: .NET Framework za korisnike i .NET Framework za developere. Ako ne programiramo neke aplikacije, ali koristimo aplikacije koje ga zahtijevaju, potrebno ga je samo instalirati na računalo. S druge strane ako

ste programer i želite koristiti ovaj framework za razvoj aplikacija, izaberite bilo koji jezik koji podržava .NET Framework za kreiranje aplikacije.

3.2.1. WINDOWS FORMS

Pod zajedničkim nazivom Windows Forms podrazumijevamo skup klasa u .NET Framework-u koje omogućavaju brzi razvoj programa s grafičkim korisničkim sučeljem. Pruža mogućnost pisanja bogatih aplikacija za tablete, prijenosna i stolna računala.

Windows Forms su besplatna biblioteka s grafičkim korisničkim sučeljem kao dio Microsoft .NET Frameworka.

Proces programiranja Windows formi je dodatno olakšan dizajnerskom podrškom razvojnog alata Visual Studio. Aplikacija većinom čeka da korisnik nešto napravi, popuni tekstualni okvir ili da pritiskom gumba aktivira određenu radnju nad podacima.

Razlikujemo dvije faze u razvoju programa:

1. vrijeme dizajniranja – vizuelno oblikovanje korisničkog sučelja pomoću Windows Forms Designer alata
2. vrijeme izvršavanja – pisanje programskog koda koji se izvršava nakon pokretanja aplikacije.

3.3. ADO.NET

U novije doba, većina aplikacija upotrebljava neki oblik pohrane podataka. Neke od renomiranih relacijskih baza podataka su danas, najčešće korišteni, tip skladišta podataka. Njihova je glavna funkcija trajna pohrana podataka koje je unio korisnik aplikacije ili podataka uvezenih iz nekog vanjskog sustava. U kontekstu web aplikacija, baze podataka služe za pohranu podataka o anketama, proizvodima, korisnicima sustava, kupcima itd. drugim riječima, bez skladišta podataka nemoguće je napraviti ozbiljnu poslovnu aplikaciju.

ADO.NET je tehnologija koja omogućuje korisnicima pristup podacima iz Microsoft .NET Framework-a i omogućuje komunikaciju između relacijskih i nerelacijskih sistema. To je određena grupa komponenti koje programeri mogu iskoristiti za pristup podacima i podatkovnim uslugama iz baze podataka. Dio je osnovne biblioteke koja je uobičajeno uključena u Microsoft .NET Framework. Najčešće se koristi za izmjenu, dodavanje i brisanje podataka u relacijskim sistemima baza podataka.

Osnovni objekti ADO.NET arhitekture odgovaraju tim operacijama i skladno tome tako su i dobili naziv Connection, Command, i DataAdapter, a DataSet i DataReader oboje pohranjuju rezultate. Postoje dvije mogućnosti kada se želimo povezati na bazu, a to su korištenje namespacea System.Data.SqlClient ili namespacea System.Data.OleDb, ovisno s kakvom bazom se želimo povezati. Ako ćemo koristiti bazu koja je napravljena preko SQL servera, onda ćemo koristiti namespace System.Data.SqlClient zbog toga što su njegovi objekti optimizirani za SQL baze, a ako ćemo koristiti baze koje su napravljene u MS Access ili Oracle programima onda ćemo koristiti namespace System.Data.OleDb. ADO.NET connection object predstavlja konekciju na bazu podataka. Sljedeće što ćemo spomenuti je Command object. Ovaj objekt služi nam kako bismo izvršavali neke radnje i naredbe nad bazom na koju smo se prethodno povezali.

4. BAZA PODATAKA - MYSQL DATABASE

Baza podataka je organizirana kolekcija podataka koji se pohranjuju i se može pristupiti preko računalnog sistema. Kada su baze podataka kompleksnije, često se izrađuju uz pomoć tehnika modeliranja i formalnog dizajna. Postoje tzv. SQL (relacijske) i NoSQL (nerelacijske) baze podataka. Relacijske baze podataka upravljaju i sadrže relacijski strukturirane podatke i imaju mogućnost manipuliranja tim podacima. Podaci se spremaju u tablice koje se sastoje od redova i stupaca. Nerelacijske baze nemaju tačnu definiciju. To je baza koja je dizajnirana za distribuciju, pristup i pohranu podataka što čini pomoću metoda, za razliku od relacijskih baza.

MySQL je open source relacijski sustav za upravljanje bazama podataka. Trenutno je instaliran na više od šest milijuna sustava i ima prosječnu stopu preuzimanje od 40.000 na dan. MySQL radi kao poslužitelj koji pruža više-korisnički pristup brojnim bazama podataka

Pored podrške za programske jezike, MySQL također ima podršku za ODBC. Ovo daje programeru mogućnost da kreira aplikacije koristeći Open Database Connectivity (ODBC) standard. Podrška za ODBC pruža mogućnost korištenja MySQL-a sa drugim menadžerima baza podataka koji imaju ODBC. Neke od tih aplikacija su Microsoft Excel, Crystal Reports, Microsoft Access i mnogi drugi. Zahvaljujući ODBC-u MySQL može se koristiti u kombinaciji sa ColdFusion-om, BorlandBuilder-om, Visual Basic-om, Delphi-em i mnogim drugim razvojnim alatima i okruženjima.

Kreiranje baze podataka **MySQL** i upravljanje sredstvima važan je dio pri kreiranju internetskih stranica pomoću posebnih platformi (CMS), kao i pri samoprogramiranju web stranice za pohranu određenih podataka: korisničkih vjerodajnica, lozinki, slika itd. **MySQL** je

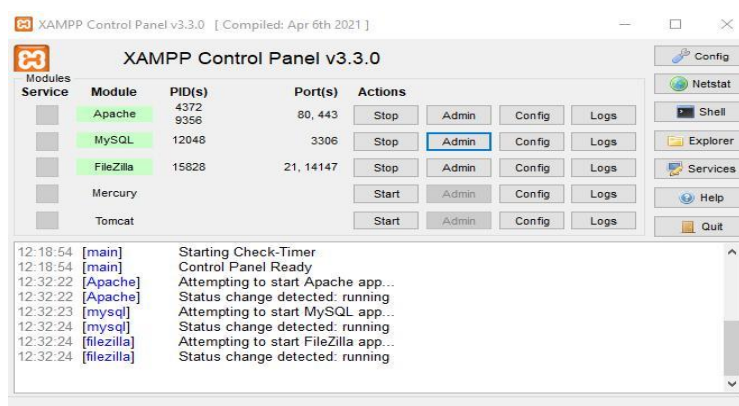
besplatan sustav za upravljanje bazom podataka otvorenog koda. Uz PostgreSQL, MySQL je čest izbor baze za projekte otvorenog koda, poput Linuxovih servera, no postoje inačice i za ostale operacijske sustave poput macOS-a, Windowsa i dr.

MySQL baze su relacijskog tipa, koji se pokazao kao najbolji način skladištenja i pretraživanja velikih količina podataka i u suštini predstavljaju osnovu svakog informacijskog sustava, tj. temelj svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija.

MySQL i PHP osvojili su veliki dio tržišta jer su otvorenog pristupa i besplatni za korištenje. MySQL je jedan od sistema za upravljanje relacionim bazama podataka. Ovaj program se ponaša kao server, sa multi-user funkcijom, odnosno dozvoljava pristup više korisnika. Svaka MySQL baza može imati nekoliko korisnika koji joj mogu pristupiti, a svaki korisnik ima predefinisane mogućnosti za rad, odnosno ovlašćenja. Ovakav pristup uz dobra podešenja, znatno umanjuje mogućnost greške. MySQL kao sistem može raditi na mnogo različitih operativnih sistema, a najčešće se koristi na Apache webserverima u kombinaciji sa PHP.

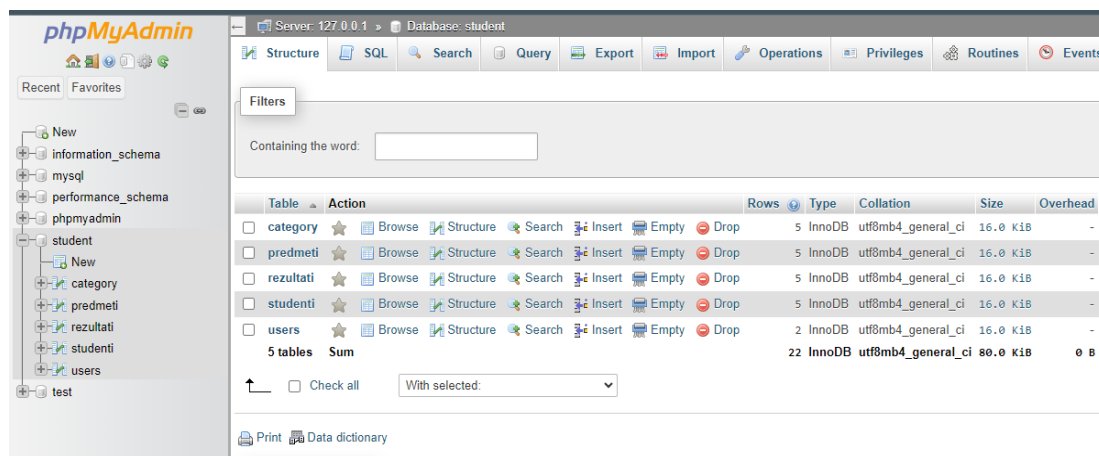
PhpMyAdmin je besplatan alat pisan u PHP-u, a služi za upravljanje i administraciju MySQL-a preko World Wide Web-a. Uz pomoć njega možete izvršavati mnoge MySQL operacije putem korisničkog sučelja izravno u Internet pregledniku. Možete upravljati bazama podataka, tablicama, poljima, indeksima, korisnicima, dozvolama pristupa samim bazama izvršavati vlastite upite na bazu i još mnogo drugih stvari, no ovo su najčešće korištene. Ono čime ću se pozabaviti u ovom tutorijalu je kreiranje same baze podataka uz pomoć phpMyAdmin web aplikacije.

Za početak morate imati pristup nekom MySQL serveru zajedno sa phpMyAdmin aplikacijom. Pod pretpostavkom da ste u ovome početnik najbolje rješenje je instalacija servera na vaše lokalno računalo, sve to možete dobiti instalacijom XAMPP Apache web servera zajedno sa MySQL i PHP-om.



Slika 1 - Izgled XAMPP aplikacije

Nakon pokretanja XAMPP Control Panela pristupamo serveru php MyAdmin upisivanjem adrese <http://localhost/phpmyadmin>, znači domena (koja je u ovom slučaju localhost ili 127.0.0.1 i /phpmyadmin. Kako bi povezali tablicu iz baze podataka koristimo neke važne metode kao što su Open() i Close(). Open() metoda omogućava nam otvaranje konekcije sa izvorom podataka koji je određen sa postavkama svojstava u connection class. Najčešće se ova metoda koristi unutar petlje „Try...Catch“ kako bismo bili sigurni da smo se povezali na bazu i da je konekcija bila uspješna. Close() metoda je usko povezana sa Open() metodom i ona se koristi kada želimo zatvoriti konekciju koju smo prije otvorili. Ona se koristi kada napravimo sve željene radnje sa podacima unutar baze i ta konekcija se mora zatvoriti inače bismo mogli dobiti neke pogreške u daljnjem radu naše aplikacije ako bismo htjeli raditi neke nove konekcije.



Slika 2 - Izgled baze podataka u phpMyAdmin

Tablica „users“ sadrži attribute: id, username, password.

Tablica „studenti“ sadrži attribute: StudentId, StudentdFirstName, StudentdLastName, StudentPhone, StudentAddress.

Tablica „predmeti“ sadrži attribute: CourseId, CourseName, CourseHour, CourseStatus.

Tablica „rezultati“ sadrži attribute: PredmetID, StudentID, NazivPredmeta, DatumIspita, Bodovi, Ocjena.

Tablica „category“ sadrži attribute: CategoryId, Label, Hours, Bodovi, Ocjena.

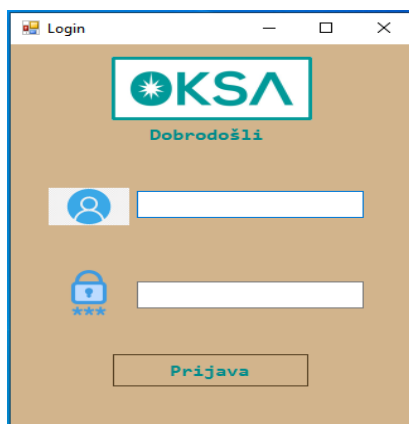
5. FUNKCIONALNOST APLIKACIJE I PRIPADAJUĆI KOD

Izgled aplikacije je zamišljen tako da se na početku otvaranja aplikacije prikazuje “SplashScreen”, koji pokazuje naziv aplikacije.

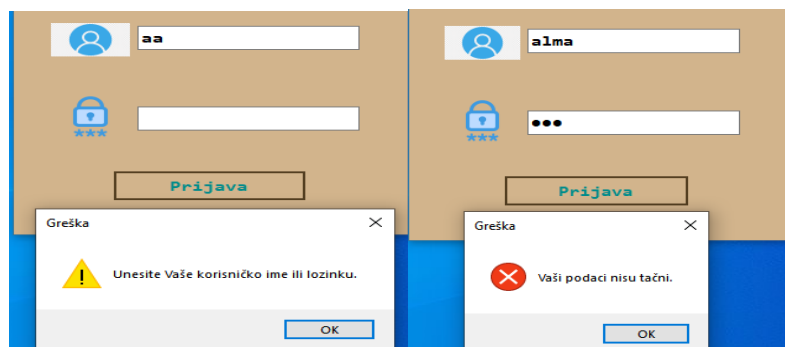


Slika 3 - SplashScreen

Nakon “SplashScreen” učitava nam se prozor “Login” koji služi za pristup aplikaciji. On sadrži Korisničko ime i Lozinku. Obzirom da je prozor povezan sa bazom podataka, aplikaciji se može pristupiti samo ako su uneseni tačni podaci. U suprotnom aplikacija će javljati greške. U nastavku su priložene slike kako to izgleda.



Slika 4 - Prijava za pristup aplikaciji



Slika 5 - Prozori obveštenja prilikom pristupa aplikaciji

Nakon uspješne prijave, učitava nam se početni prozor koji je prilično jednostavan za korištenje, uz to dodan je i gumb za izlaz iz aplikacije. Odabir na neku od opcija otvaraju se novi prozori uz pomoć “Show()” metode.



Slika 6 - Početna stranica (meni) aplikacije

Odabirom “Student” iskače nam forma koja sadrži podatke o studentima. U ovom prozoru imamo opcije poput dodavanja novog studenta u bazu podataka; uređivanje podataka o trenutnim studentima; kao i potpuno brisanje studenata iz baze podataka. Nakon što se popune svi tekstualni okviri, pritisne se gumb „Sačuvaj“/”Ažuriraj” i podaci se spremaju u bazu podataka, tačnije u tablicu „studenti“. Nakon unešenih podataka i pritiskom na gumb “Sačuvaj“, podaci se ispisuju u DataGridView, na osnovu toga student je uspješno sačuvan.

Potrebno je naglasiti da su svi podaci potrebni za izradu baze podataka studenata imaginarnog tipa, i služili su samo za razvoj aplikacije.

OKSA **STUDENTI**

181	Ensar	Šarić	062/035/983	Zenica
36	Lajla	Tukić	062/711/011	Mostar
4	Amina	Tarčin	062/554/025	Sarajevo
5	Erol	Lendić	061/889/902	Tuzla
88	Emir	Konjić	063/784/002	Srebrenik

Početna
Student
Predmet
Rezultati
Izlaz

ID: Ime: Prezime:
Kontakt broj: Adresa:

Sačuvaj **Ažuriraj** **Izbriši** **Očisti**

Slika 7 - Izgled izbornika "Student"

88 Emir Konjić 063/784/002 Srebrenik

Greška
Molimo unesite tražene podatke.
OK

ID: Ime: Prezime:
Kontakt broj: Adresa:

Sačuvaj **Ažuriraj** **Izbriši** **Očisti**

Slika 8 - Prilikom unosa podataka za evidenciju studenta, obavezno je unijeti sve potrebne podatke. U protivnom student neće biti sačuvan.

STUDENTI

181	Ensar	Šarić	062/035/983	Zenica
36	Lajla	Tukić	062/711/011	Mostar
4	Amina	Tarčin	062/554/025	Sarajevo
5	Erol	Lendić	061/889/902	Tuzla
88	Emir	Konjić	063/784/002	Srebrenik

Spremljeni podaci
Podaci o studentu su uspešno spremljeni.
OK

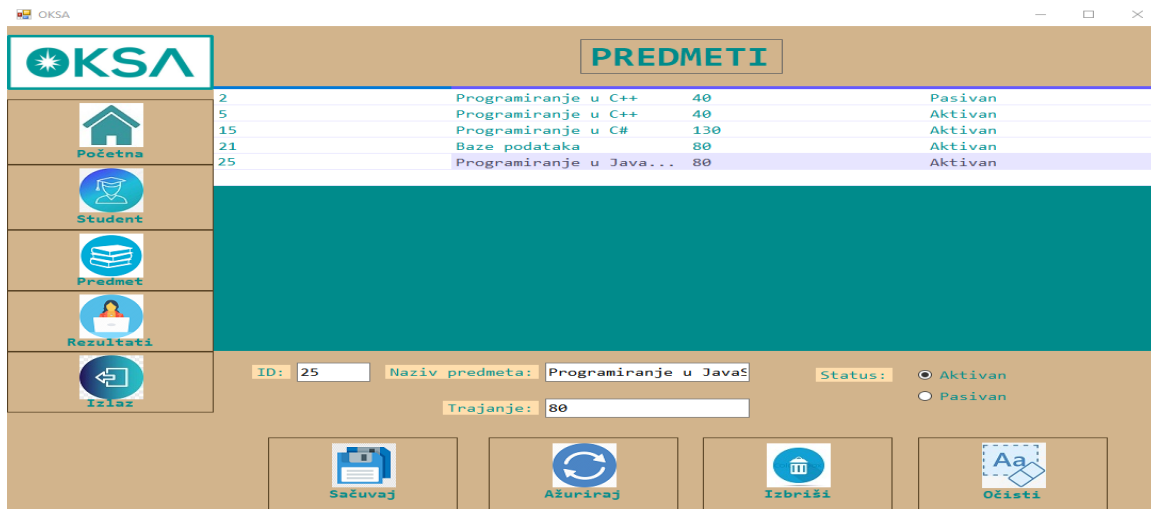
ID: Ime: Prezime:
Kontakt broj: Adresa:

STUDENTI

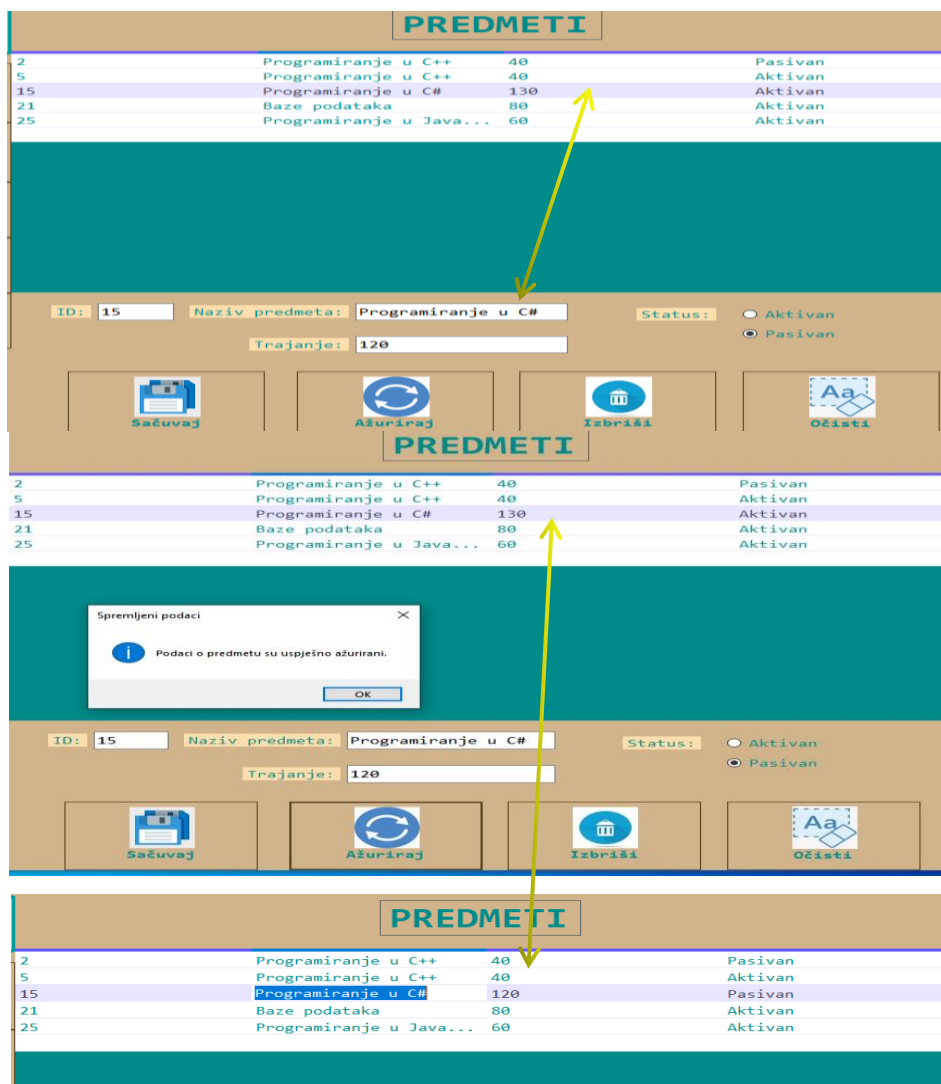
181	Ensar	Šarić	062/035/983	Zenica
26	Alma	Šabić	061568939	Donje Dubrave
36	Lajla	Tukić	062/711/011	Mostar
4	Amina	Tarčin	062/554/025	Sarajevo

Slika 9 - Uspješno evidentiran novi student.

Odabirom “Predmet” iskače nam forma koja sadrži podatke o predmetima. U ovom prozoru imamo opcije poput dodavanja novog predmeta u bazu podataka; uređivanje podataka o trenutnim predmetima; kao i potpuno brisanje predmeta iz baze podataka.



Slika 10 - Prikaz izbornika “Predmeti”



Slika 11 - Ažuriranje podataka o predmetu

Odabirom “Rezultati” iskače nam forma koja sadrži podatke o postignućima studenata iz predmeta. U ovom prozoru imamo opcije poput dodavanja novog termina rađenog ispita u bazu podataka; unos bodova kao i ocjene po predmetima.

ID	Predmet	Bodovi	Ocjena
2	Programiranje u...	40	82
5	Programiranje u...	40	42
15	Programiranje u C#	130	65
21	Baze podataka	80	80
25	Programiranje u...	80	50

Forma za unos podataka:

PredmetID: Naziv predmeta: Programiranje u C++ Bodovi:

StudentID: Datum ispita: 5/30/2023 Ocjena: 8

Buttons: Sačuvaj, Očisti

Slika 12 - Prikaz izbornika "Rezultati"

Spremljeni podaci

Rezultati su uspešno spremljeni.

OK

Forma za unos podataka:

PredmetID: Naziv predmeta: Baze podataka Bodovi: 72

StudentID: 26 Datum ispita: 5/23/2023 Ocjena: 7

Buttons: Sačuvaj, Očisti

Slika 13 - Spremanje novih rezultata

6. ZAKLJUČAK

Aplikacija sadrži različite mogućnosti. Iz aspekta profesora, funkcionalnost je jednostavna za korištenje koja korisniku omogućuje uvid u stanje o studentu. Aplikacija je osmišljena na taj način da se na jednom mjestu nalaze svi podaci o studentima. Tako možemo pronaći na aplikaciji osnovne podatke o studentu, predmete koje on pohađa kao i ocjene iz tih predmeta.

Kako u suvremeno doba manje više svi koriste računala ideja za aplikaciju proizašla je iz istog razloga kao i želje za napretkom u ovom području IT industrije. Odabir korištene tehnologije proizašao je iz dosadašnjeg iskustva na kolegijima na fakultetu te samom popularnošću jezika uz cilj unaprijeđenja znanja iz tog područja. Osim što sam unaprijedila dosadašnja znanja, korištenjem navedenih alata i metoda susrela sam se sa novim tehnologijama i iskustvima koje će mi pomoći u daljnjem razvoju. Nakon izrade praktičnog rješenja puno toga je sjelo na svoje mjesto te smatram da bi svaki idući proizvod bio za nijansu bolji od prethodnoga. Kroz ovaj rad naučeni su osnovni koncepti i pokriveni su neki elementi koje bi, manje više, svaka aplikacija trebala imati. Iako je dosta pokriveno unutar rada, okruženje „Microsoft Visual Studio“ nam pruža puno drugih mogućnosti te se uvijek može otkriti nešto novo.

Što se tiče svrhe aplikacije, njen cilj je olakšati život njenim korisnicima. Mislim da se u tom cilju uspjelo. Kao i svaka druga aplikacija, i ova se može nadograđivati, uređivati u skladu sa potrebama korisnika. Za potrebe nadogradnje aplikacije, potrebno je da se još više uđe u dodir sa programskim jezikom C#, kao i sa svim mogućnostima koje pruža Visual Studio.

Za daljnje istraživanje savjetujem da se poradi na proučavanju svih mogućnosti koje nam pružaju programski jezik C# i integrirano razvojno okruženje Visual Studio.

7. LITERATURA

- Ćirović, Z. i Dunderski, I. (2006). *Tehnike vizuelnog programiranja u C#*. Beograd: Viša elektrotehnička škola.
- Davila, S. i Kaštelan, T. (2010). *Osnove SQL-a*. Zagreb: Algebra.
- Janssen, D. *Relational model*, Technopedia: <https://www.techopedia.com/definition/24559/relational-model-database> [Pristup 05.06.2023.]
- Vodanović, A.. i Pavić, I. (2017). *MVC razvoj aplikacija: Povezivanje s bazama podataka*. Zagreb: Algebra.
- Köhler, M. i Lukić, I.. (2011). *Osnove Internet programiranja*. Osijek: Elektrotehnički fakultet Osijek, Sveučilište J.J. Strossmayera u Osijeku.
- Kaštelan, T. i Mesić, I. (2009). *Uvod u baze podataka*. Zagreb: Algebra.
- Matić, D. (1996). *Objektno - orijentisano programiranje u jeziku C#*. Beograd.
- Millet, S. (2010). *Professional ASP.NET Design Patterns*. Indianapolis: Wiley Publishing.

Alma Okić Šabić, M. Sc.

DEVELOPMENT OF THE APPLICATION FOR STUDENT RECORDS

SUMMARY: The entire work is based on the development of a C# application and its connection with a database, in this case with MySQL databases. C# is a programming language used to create applications that can run under the .NET environment. Visual Studio .NET was used for programming in C#, which greatly facilitated the development of the application itself. In addition to containing application design tools, tools for displaying and navigating project elements, Visual Studio .NET enables easier project distribution and customization to clients, and provides advanced project development debugging techniques. C# is a fully object-oriented programming language, which means that all elements within it represent an object. A MySQL database has been created in which we will find the necessary data, and which will be accessible through the application. MySQL is a free, open source database management system. The idea and goal of the application was to provide an efficient and reliable management database for professional needs.

KEYWORDS: C#, programming language, MySQL, database, Visual Studio