**Прва београдска гимназија**

Београд, Цара Душана 61

МАТУРСКИ РАД ИЗ ВЕБ ПРОГРАМИРАЊА

ВЕБ АПЛИКАЦИЈА ЗА КОЛЕКЦИОНАРЕ СЛИЧИЦА

Ментор: Ученик:

Александар Герасимовић, *проф.*  Андреј Братић

Београд, мај 2022. год.

**С А Д Р Ж А Ј**

**1 УВОД2**

**2 БЕКЕНД4**

2.1 КРЕИРАЊЕ БАЗЕ4

2.2 ПРОЦЕДУРЕ6

**3 ФРОНТЕНД8**

3.1 КЛАСЕ ЗА ИМПЛЕМЕНТАЦИЈУ ПРОЦЕДУРА 8

3.2 ГЛАВНЕ СТРАНИЦЕ АПЛИКАЦИЈЕ 9

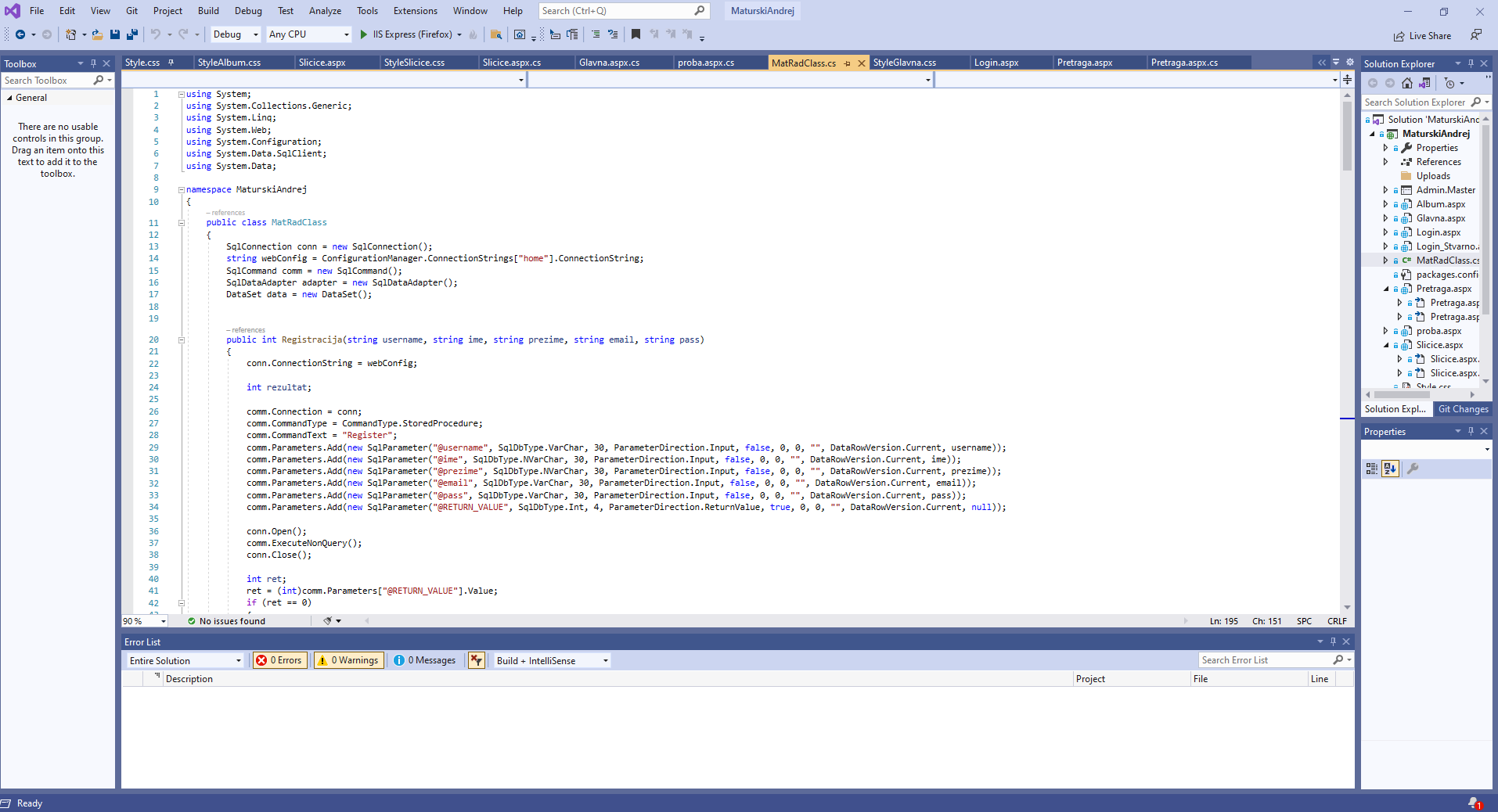
**4 ЛИТЕРАТУРА16**

# 1

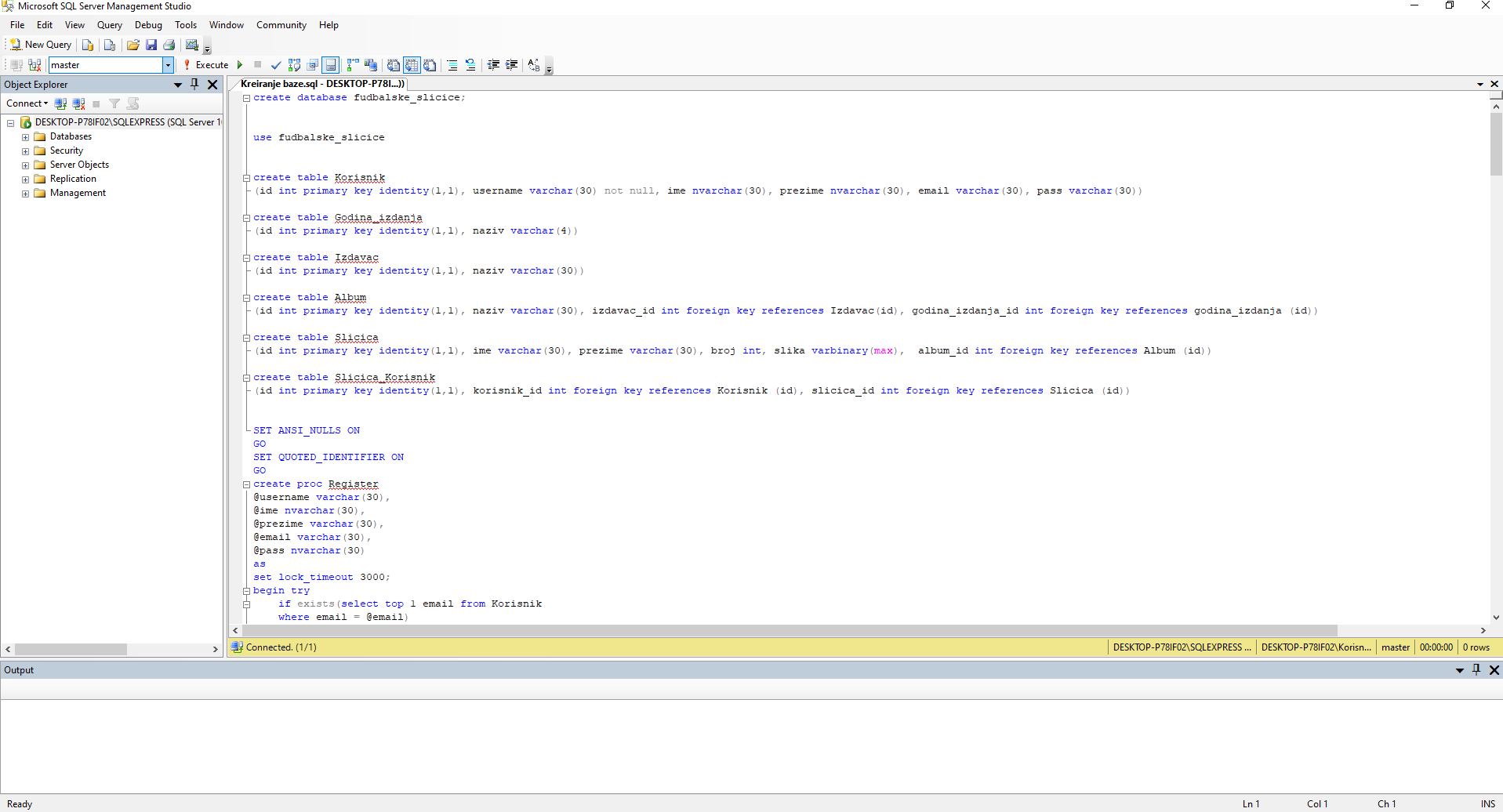
# **УВОД**

Циљ веб апликације за колекционаре сличица је да омогући колекционарима да пронађу сличице које им требају помоћу интерфејса који је представљен корисницима. Корисници могу да додају сличице на сајт као и да претражују сличице других корисника. Сајт је примарно намењен за колекционаре фудбалских сличица.

За израду веб апликације коришћена је ,,ASP.NET” технологија. Изабрао сам ову технологиију зато што је погодна за израду веб форми које су биле потребне за реализацију овог пројекта, као и зато што је једна од масовније кориштених технологија. Коришћен је програмски језик C# (си-шарп) у апликацији ,,Microsoft Visual Studio 2019”. Што се тиче базе података коришћен је језик SQL у апликацији ,,SQL Server Management Studio”. За изглед странице и кориснички интерфејс коришћени су HTML, CSS и JavaScript.



Слика 1.1 : Microsoft Visual Studio 2019



Слика 1.2: SQL Server Management Studio

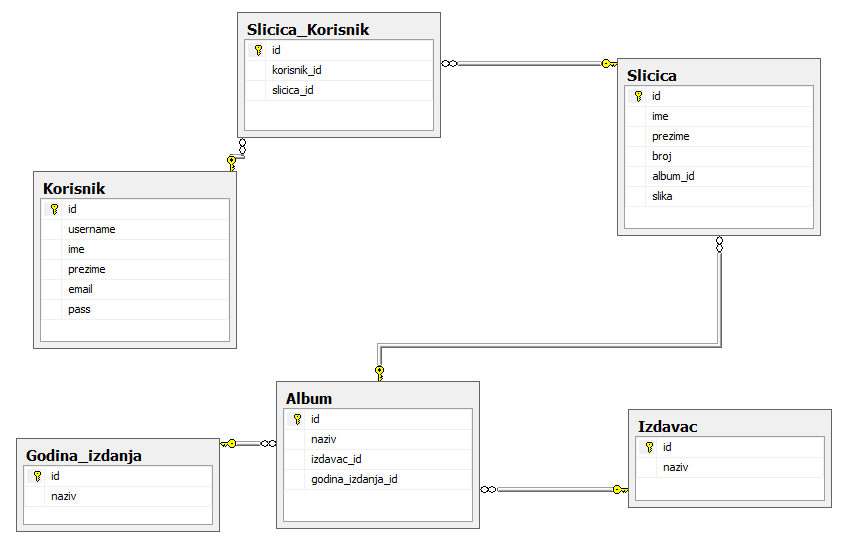
# 2

# **БЕКЕНД**

# Креирање базе

Бекенд ове апликације се састоји из табела у којима складиштимо податке и процедура којима вршимо неке измене над тим табелама. Табеле направљене за ову апликацију су:

* Табела ,,Корисник” – у њој складиштимо податке о кориснику који приступа апликацији
* Табела ,,Албум” – у њој складиптимо податке о албумима сличица које постоје у нашој бази података
* Табела ,,Сличица” – у њој складиштимо основне податке о сличици коју је корисник додао
* Табела ,,Сличица\_Корисник” – у њој складиштимо информацију о томе који корисник је власник које сличице
* Табела ,,Година\_Издања – у њој складиштимо списак година издања албума у бази података
* Табела ,,Издавачи” – у њој складиштимо списак издавача албума у бази података



Слика 2.1: Дијаграм базе података

Следи код за конструисање ове базе података:

create database fudbalske\_slicice;

use fudbalske\_slicice

create table Korisnik

(id int primary key identity(1,1), username varchar(30) not null, ime nvarchar(30), prezime nvarchar(30), email varchar(30), pass varchar(30))

create table Godina\_izdanja

(id int primary key identity(1,1), naziv varchar(4))

create table Izdavac

(id int primary key identity(1,1), naziv varchar(30))

create table Album

(id int primary key identity(1,1), naziv varchar(30), izdavac\_id int foreign key references Izdavac(id), godina\_izdanja\_id int foreign key references godina\_izdanja (id))

create table Slicica

(id int primary key identity(1,1), ime varchar(30), prezime varchar(30), broj int, slika varbinary(max), album\_id int foreign key references Album (id))

create table Slicica\_Korisnik

(id int primary key identity(1,1), korisnik\_id int foreign key references Korisnik (id), slicica\_id int foreign key references Slicica (id))

alter table Slicica

add slika nvarchar(max)

# Процедуре

Сада ћу навести код процедура кориштених за рад над креираним табелама.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create proc Register

@username varchar(30),

@ime nvarchar(30),

@prezime varchar(30),

@email varchar(30),

@pass nvarchar(30)

as

set lock\_timeout 3000;

begin try

if exists(select top 1 email from Korisnik

where email = @email)

return 1

else

insert into Korisnik(username,ime,prezime,email,pass)

values (@username,@ime,@prezime,@email,@pass)

return 0;

end try

begin catch

return @@error

end catch

Create procedure Korisnik\_Provera

@email varchar(30),

@pass varchar(30)

AS

SET LOCK\_TIMEOUT 3000;

BEGIN TRY

IF EXISTS(SELECT TOP 1 email FROM Korisnik

WHERE email = @email and pass=@pass)

Begin

RETURN 0

end

RETURN 1

END TRY

BEGIN CATCH

RETURN @@error;

END CATCH

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure Godina\_Izdanja\_Insert

@godina varchar(4)

as

set lock\_timeout 3000;

begin try

if exists(select top 1 naziv from Godina\_izdanja

where naziv=@godina)

return 1

else

insert into Godina\_izdanja(naziv)

values (@godina)

return 0;

end try

begin catch

return @@error

end catch

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

alter procedure Album\_Insert

@naziv varchar(30),

@izdavac\_id int,

@godina\_izdanja int

as

set lock\_timeout 3000;

begin try

if exists(select top 1 naziv from Album

where naziv=@naziv and izdavac\_id=@izdavac\_id and godina\_izdanja\_id=@godina\_izdanja)

return @izdavac\_id

else

insert into Album(naziv,izdavac\_id,godina\_izdanja\_id)

values (@naziv,@izdavac\_id,@godina\_izdanja)

return 0;

end try

begin catch

return @@error

end catch

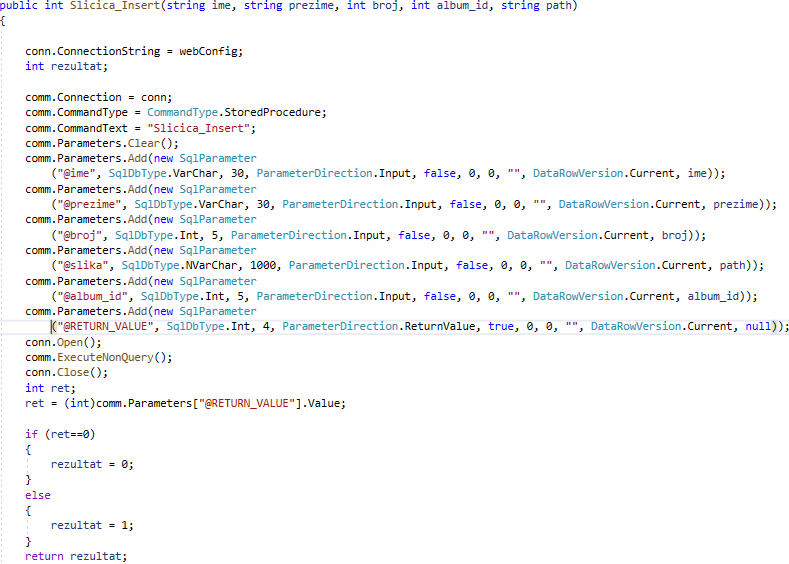
По овом шаблону урађен је остатак процедура за остале табеле.

# 3

**Фронтенд**

# Класа за импленетацију процедура

Класа ,,MatRadClass.cs” је коришћена ради прављења метода које ће позивати процедуре за попуњавање табела које смо направили у бекенду. Сада ћу приказати пример кода из ,,Microsoft Visual Studio 2019”.



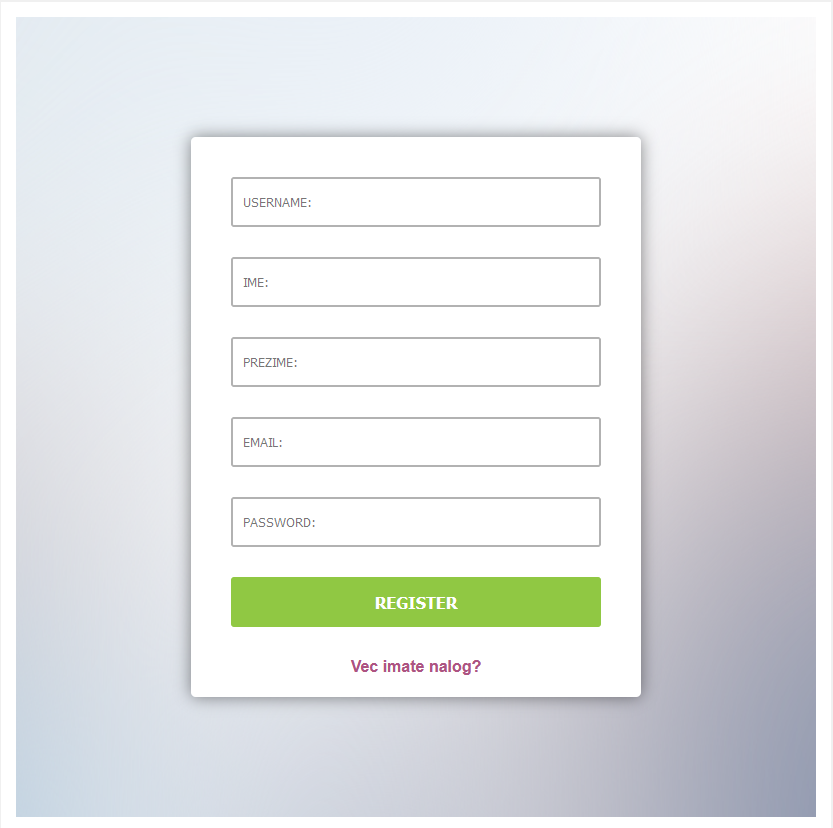
3.1 слика: Пример методе за имплементацију SQL процедура

# Странице за регистрацију и пријављивање

Када корисник учита апликацију прво што ће га дочекати јесте страница за пријављивање. У случају да нема налог, може да оде на страницу за регистрацију.



3.2 слика: Страница за пријављивање



3.3 слика: Страница за регистрацију

Уколико је унета било каква врста неисправних података апликација о томе обавести корисника. Када је установљено да су унесени добри подаци за корисника, апликација нас шаље на следећу страницу.

## Главне странице апликације

Прва страница која ће бити приказана је нека врста навигације. На њој могу да се одаберу три странице на које даље можемо да идемо, које уједно и чине главну идеју ове апликације.



Слика 3.4: Страница за навигацију

Уколико одаберемо линк ,,DODAJ ALBUM” приказаће нам се страница на којој можемо да видемо доступне албуме у нашој бази података, а такође можемо да додамо и нови албум, да када будемо додавали нову сличицу, можемо да одаберемо одговарајући албум.

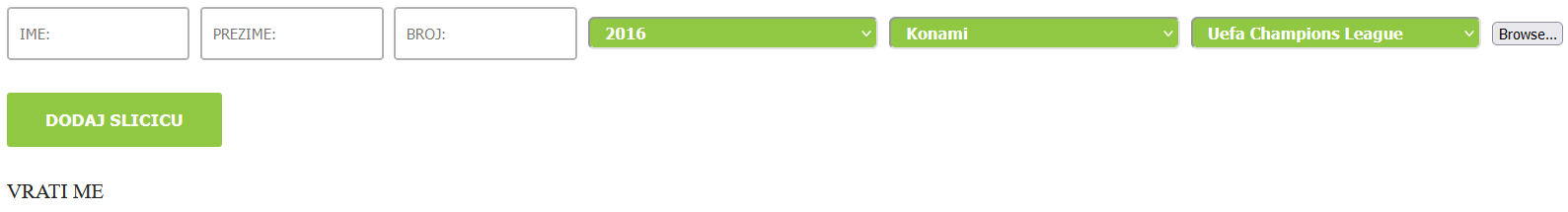


Слика 3.5: Страница за додавање албума

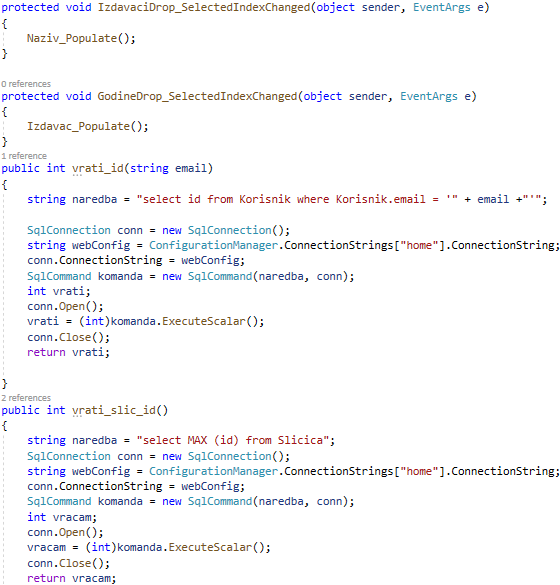


Слика 3.6: Део кода за прављење странице за додавање албума

Следећи линк на навигационој страни, ,,DODAJ SLICICE” нас води на страницу на којој корисник може да дода сличицу коју поседује у нашу базу, како би је касније други корисници могли видети. Корисник чак има и опцију да приложи слику уз информације о сличици.



Слика 3.7: Страница за додавање сличица

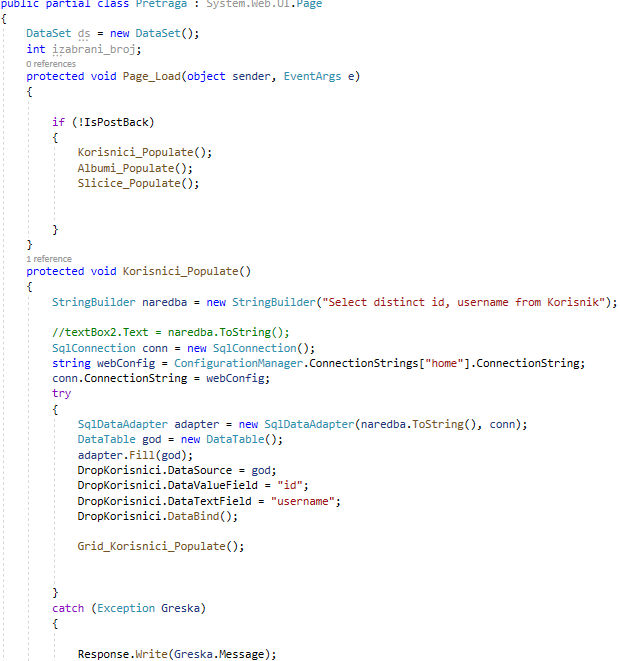


Слика 3.8: Део кода за прављење странице за додавање сличица

Последњи линк на навигационој страници ,,PRETRAZI SLICICE” води до странице која служи за преглед садржаја које ова апликација нуди. Може да да извештај о томе да ли постоји сличица, ко је њен власник и да ли постоји нека сличица из одговарајућег албума. Све те информације могу да се добију уз одговарајуће одабране параметре које апликација сама нуди.



Слика 3.9: Страница за претрагу базе података апликације



Слика 3.10: Део кода за прављење странице з а претрагу базе података апликације

Закључно са овом страницом сумиране су све фунцкионалности веб апликације за колекционаре сличица.

# 4

**Литература**

* <https://www.c-sharpcorner.com>
* <https://www.w3schools.com/>
* <https://classroom.google.com/u/1/c/MzE5OTQwMDY2NjM1>

Датум предаје: \_\_\_\_\_\_\_\_\_\_\_\_\_

Комисија:

Председник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Испитивач \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члан \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Коментар:

Датум одбране: \_\_\_\_\_\_\_\_\_\_\_\_ Писмени испит \_\_\_\_\_\_\_\_\_\_(\_\_\_)

Усмени испит \_\_\_\_\_\_\_\_\_\_(\_\_\_)

Закључна оцена \_\_\_\_\_\_\_\_\_\_(\_\_\_)