**Универзитет ,,Гоце Делчев” – Штип**

**Факултет за Информатика**

**Насока: Компјутерско инженерство и технологии**

**Предмет: Објектно ориентирано програмирање**

A circular logo with a building in the middle

Description automatically generated

**Дипломски труд:**

**Апликација за центар за обуки за професионално усовршување**

**Ментор: Кандидат:**

**проф. д-р Цвета Мартиновска Банде Андреј Стоилов**

**102193**

**Штип, Октомври 2023**

**Содржина**

[1.1. Апстракт 4](#_Toc146958071)

[2. Вовед 5](#_Toc146958072)

[2.1. Цел на овој документ 6](#_Toc146958073)

[2.2. Мотивација за изработка на проектот 6](#_Toc146958074)

[2.3. Краток опис на проектот 6](#_Toc146958075)

[3. Дизајн процес 9](#_Toc146958076)

[3.1. Вовед во Microsoft Visual Studio 10](#_Toc146958077)

[Организација и структура на апликацијата 14](#_Toc146958078)

[4. Најава во апликацијата (Login) 14](#_Toc146958079)

[4.1. Почетна страна (Home Page) 15](#_Toc146958080)

[4.2. Мени за менаџмент со клиенти 15](#_Toc146958081)

[4.3 Преглед и внес на присутство 16](#_Toc146958082)

[4.4 Генерирање и плаќање на трошок на клиент 18](#_Toc146958083)

[4.5 Мени за менаџмент со вработени 19](#_Toc146958084)

[4.6 Исплаќање на вработен 20](#_Toc146958085)

[4.7 Мени за преглед и внес на трошоци 21](#_Toc146958086)

[4.8 Мени за финансиски извештаи 22](#_Toc146958087)

[4.8.1 Извештаи за клиенти 22](#_Toc146958088)

[4.8.2 Извештаи за приход 23](#_Toc146958089)

[4.8.3 Извештаи за неплатени трошоци 24](#_Toc146958090)

[4.8.4 Извештаи за месечни плати 25](#_Toc146958091)

[4.9 Опции 26](#_Toc146958092)

[4.10 Мени за промена на информации при најавување 27](#_Toc146958093)

[4.11 Мени за Back-Up и Restore на базата на податоци 28](#_Toc146958094)

[5 Структура на датотеки на апликацијата 29](#_Toc146958095)

[6 Front end 30](#_Toc146958096)

[6.1 C# 30](#_Toc146958097)

[7 Back end 31](#_Toc146958098)

[7.1 Фолдер App\_Code 31](#_Toc146958099)

[7.2 Систем за автентификација со AES 33](#_Toc146958100)

[7.3 Шифрирање на лозинки 34](#_Toc146958101)

[8 Креирање на базата на податоци 35](#_Toc146958102)

[8.1 ERD(Entity relationship diagram) дијаграм 35](#_Toc146958103)

[8.2 Структура на табелите во Microsoft SQL Server Management Studio 38](#_Toc146958104)

[9 Заклучок 43](#_Toc146958105)

[10 Користена литература 44](#_Toc146958106)

## Апстракт

Апликацијата за управување со центар за обука и професионално усовршување е сеопфатно софтверско решение развиено со користење на програмскиот јазик C#, дизајнирано да ги револуционизира и рационализира административните, академските и оперативните процеси во овие институции. Во време кое се карактеризира со брза дигитална трансформација, традиционалните рачни методи за управување со евиденција на клиенти, информациите за институцијата, распоредот на курсеви и финансиските трансакции станаа несоодветни. Овој проект има за цел да ги реши овие проблеми преку обезбедување на лесен, приспособлив и скалабилен систем кој им овозможува на центрите за обука да ги автоматизираат нивните секојдневни операции. Системот за управување со обуки за професионално усовршување нуди модули за управување со информации за клиентите, детали за центарот за обука, преглед на курсеви и предмети, финансиски трансакции, преглед и менаџмент со вработени, преглед на дневни, месечни и годишни извештаи. Овие модули колективно ја подобруваат ефикасноста, точноста и транспарентноста. Клучните карактеристики вклучуваат менаџмент за уписи на клиенти и нивни информации, информации и контакт со центарот за обука, преглед на курсеви, финансиско следење, менаџмент на вработените, извештаи и преглед на сите трошоци што една институција може да ги има. Овој проект не само што ги поедноставува административните задачи, туку и поттикнува донесување одлуки водени од податоци во овие институции. Најголема предност е нудењето на приспособливост и адаптација за да одговараат на институциите од сите големини. Апликацијата за управување со центри за обука и професионално усовршување претставува возбудлива можност за искористување на технологијата за подобрување на администрацијата во овие институции. Тој предвидува иднина каде што овие институции ќе можат да ги оптимизираат своите ресурси и да обезбедат супериорно образовно искуство за клиентите. Овој систем е доказ за моќта на технологијата во подобрувањето на ефикасноста и ефективноста на образовниот менаџмент.

Apstract

**The application for management of courses and professional improvement** is a comprehensive software solution developed using the C# programming language, designed to revolutionize and streamline the administrative, academic, and operational processes within educational institutions. In an era characterized by rapid digital transformation, traditional manual methods of managing client records, company information, course scheduling, and financial transactions have become inadequate. This project aims to bridge this gap by providing a user-friendly, adaptable, and scalable system that empowers institutions like this to automate their daily operations. This system offers modules for managing client information, company details, course scheduling, financial transactions, employee management, daily, monthly and yearly financial reports. These modules collectively enhance efficiency, accuracy, and transparency, allowing educational institutions to focus on their core mission of delivering quality education. Key features include client record management, company details, course view, financial tracking, digital records of every transaction, tracking attendance. Additionally, it aims to help in making informed decisions and improve overall institutional performance. This project not only simplifies administrative tasks but also fosters a culture of data-driven decision-making within educational institutions. It addresses the evolving needs of professional improvement institutions by offering scalability and adaptability to suit institutions of all sizes. This system represents an exciting opportunity to harness technology for the betterment of education administration. It envisions a future where educational institutions can optimize their resources and provide a superior educational experience to clients.

# Вовед

Во постојано еволуирачката област на образованието, управувањето со административни задачи, академски записи и оперативна логистика центрите за обука се постојан предизвик. За да се справиме со овој предизвик, е осмислена апликацијата за управување со центар за обуки. Ова копмлексно софтверско решение, креирано со користење на C# програмскиот јазик, е спремно да го преосмисли начинот на кој овие институции се справуваат со нивните ресурси, податоци и интеракции. Во денешната дигитална ера, традиционалните методи за управување со записите на студентите, информациите за професорите, распоредот на предавања и финансиските трансакции веќе не се доволни. Оваа апликација за менаџмент нуди целосен пристап за да се совладаат овие предизвици. Тој овозможува на факултетите и универзитетите да ги автоматизираат и оптимизираат своите дневни операции, дозволувајќи им да се фокусираат повеќе на обезбедувањето на висококвалитетно образование, а помалку на административните потешкотии. Овој проект претставува уникатна можност за искористување на потенцијалот на технологијата за да се задоволат специфичните потреби на образовниот сектор. Тоа ветува да ги поедностави комплексностите поврзани со водењето на образовните институции и да им овозможи да напредуваат во дигиталната ера. Овој проект, ги поедноставува операциите, ја зголемува точноста и ја подобрува комуникацијата меѓу студентите, професорите и администрацијата. Со обезбедување на информации базирани на податоци и олеснување на ефикасното управување, апликацијата за менаџмент со центри за обука и професионално усовршување е спремна да го преобрази начинот на кој овие институции функционираат.

## Цел на овој документ

Во овој дипломски труд е опишан целиот процес на создавање на апликацијата за центар за обуки за професионално усовршување во кои е вклучено Front-End програмирањето и Back-End програмирањето, заедно со креирањето и структурата на базата на податоци.

## Мотивација за изработка на проектот

Инспирација за оваа апликација добив од е-индекс системот на Универзитетот „Гоце Делчев“. Апликацијата може лесно да се имплементира во било кој центар за обука, независно од насоката или предметите што се учат. Имплементацијата е брза и лесна и дава огромен преглед на корисникот над клиентите, вработените, нивните трошоци и плати, како и присутството на клиентите.

## Краток опис на проектот

Главна цел е да се постигне голема функционалност на апликацијата. За да се реализира тоа потребно е:

1. **Менаџмент на клиенти**- Прво мени на апликацијата е менито за преглед, додавање и бришење на клиенти. Во ова мени може да се видат сите моментални клиенти на центарот за обуки, да се додаде клиент на било кој предмет или насока каде што се внесуваат лични податоци за клиентот како име презиме, адреса, емаил, дата на раѓање. Исто така може и целосно да се избрише одреден клиент.
2. **Генерирање на трошок, плаќање на трошок-** Во менито за клиенти имаме опција за еден клиент да генерираме трошок кој може да е за полагање на испит, упис на одреден курс и плакање на истиот.
3. **Внес и преглед на присутство**- За секој клиент е дадена опцијата да се води евиденција за неговото присутство на еден курс, пристап до ова има администраторот на курсот, а истото може и да се прегледа како месечно присутство на одреден клиент.
4. **Принтање на профил**- Опцијата за преглед и принтање на еден профил е достапна и за клиентот и за вработените. Овде може да се видат сите информации за еден клиент и вработен и истите да се испечатат или зачуваат како PDF, Word или Excel документ.
5. **Менаџмент со вработени-** Во менито за вработени може да се додаваат бришат и променуваат продатоци за даден вработен. Во ова мени се имплементирани и опциите за исплата на плата, преглед и додавање на квалификации на вработениот кои подоцна се покажуваат на неговиот профил.
6. **Трошоци-** Во менито за трошоци може да се внесат дневни трошоци кои ги имаат вработените во центарот за обука, кои подоцна можат да се прегледаат како вкупно за цел месец, или како цел извештај.
7. **Извештај-** Во менито за извештаи е овозможен преглед на финансиите на центарот за обука. Тука може да се прегледаат: дневни, месечни и годишни трошоци на центарот за обука, клиенти кои се стипендирани како и клиенти кои се на ист курс. Додадени се опциите за преглед на годишни приходи на центарот за обука каде се гледа извештај за сите платени трошоци од страна на клиентите, како и извештај за неплатените. Исто така е овозможен месечен преглед на исплата на вработените.
8. **Опции-** Ова е голем дел од олеснувањето при имплементација на апликацијата во било кој центар за обуки. Администраторот може да ги внесе името, цените, мото на фирмата, како и контакт информациите и логото за центарот на обуки. Сето ова влегува во базата на податоци и игра голема улога во функционалноста на апликацијата.

При самото вклучување апликацијата бара да внесеме наши податоци или login информации.

**Регистрирани корисници**- При внесувањето на вработен, имаме опција да селектираме систем улога, дали тоа бил систем администратор или не. Систем администраторите ги имаат привилегиите да менуваат, додаваат и бришат информации, додека обичниот корисик може само да има преглед на информациите.

**Регистрација на корисници**- Регистрацијата се врши при додавање на вработен во системот и доделување на привилегии.

Технологии користени во апликацијата

**Управување со База на податоци/Database management:** Microsoft SQL Server Management Studio(бесплатна интегрирана средина за менаџмент со било која SQL инфраструктура. Нуди пристап, менаџмент и развој на сите компоненти на SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server на Azure VM, и Azure Synapse Analytics ).

Проектот е целосно изработен во C# вклучувајќи ги и front-end делот и back-end делот.

**База на податоци/Database:** MySQL

**Code editor:** Microsoft Visual Studio

**Оперативен систем/Operating system:** Windows 10/ Windows 11

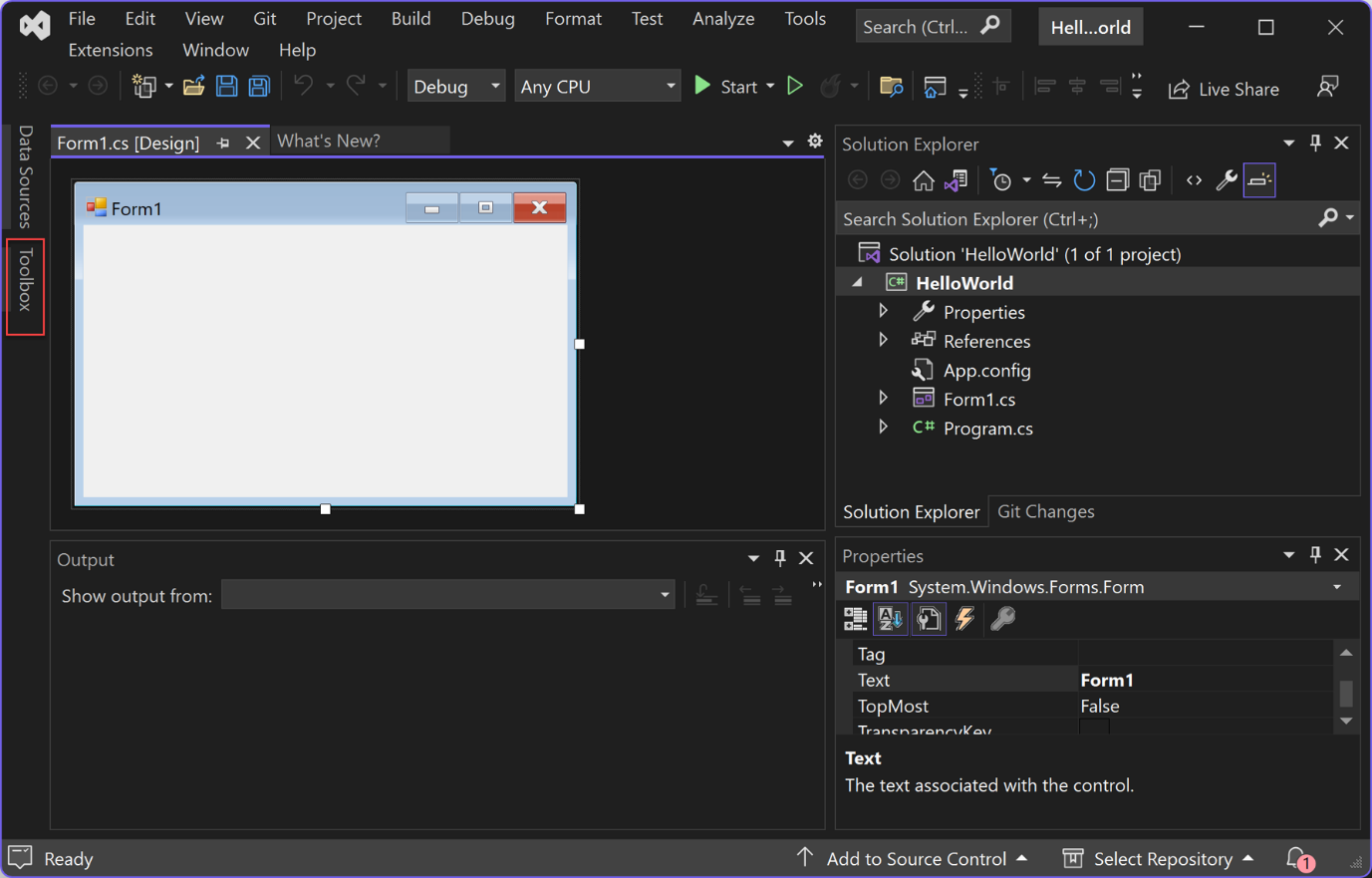
**Програма за изработка на дизајн/Design program:** Microsoft Visual Studio

**Создавање на дијаграми/Diagram design:** Visual Paradigm Online ([https://online.visualhttps://online.visual-paradigm.com/paradigm.com)](https://online.visual-paradigm.com/)

# Дизајн процес

Првиот чекор од создавањето на апликацијата е фазата на истражување на конкуренцијата на пазарот. Доколку има добра проценка за апликацијата да успее, се преминува на фазата на дизајнирање на екраните(страните) на апликацијата.

За изработка на дизајнот, е користен Microsoft Visual Studio.



1. Интерфејс на Microsoft Visual Studio

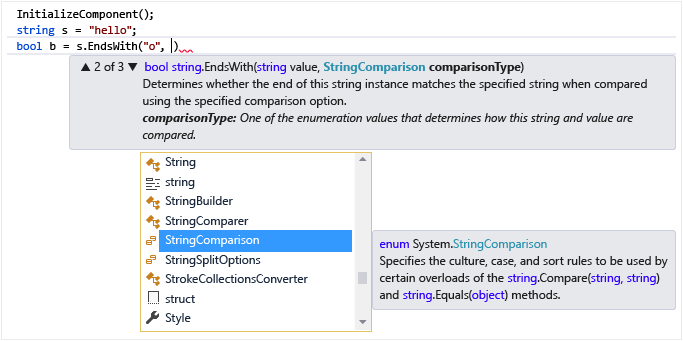
## Вовед во Microsoft Visual Studio

Visual Studio е моќна алатка за развивачи што можете да ја користите за да го завршите целиот циклус на развој на едно место. Тоа е сеопфатна интегрирана развојна околина (IDE) што можете да ја користите за пишување, уредување, дебагирање и создавање код, а потоа распоредување на вашата апликација. Надвор од уредувањето и дебагирањето на кодот, Visual Studio вклучува компајлери, алатки за комплетирање код, контрола на изворот, екстензии и многу повеќе функции за подобрување на секоја фаза од процесот на развој на софтвер.

1. **Windows Forms Designer**- се користи за дизајн и изградба на GUI (Graphical User Interface) или графички кориснички интерфејс.

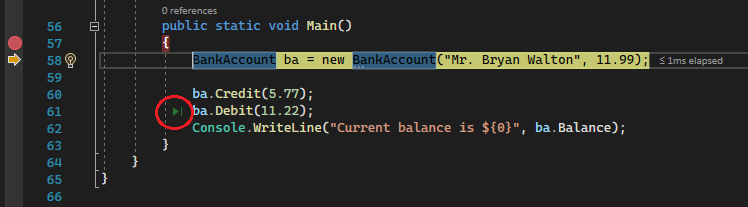


1. Интерфејс на Windows Forms
2. **Code Editor**- пишување и модификација на код во голем број на моментално актуелните програмски јазици како што се C++, Java, Javascript, CSS, C#, Pyhton. Едиторот на Visual Studio подржува и алатка наречена IntelliSense која што има функција за истакнување, исправка и дополнување на код, што го прави Visual Studio еден од најкористените IDE денес.



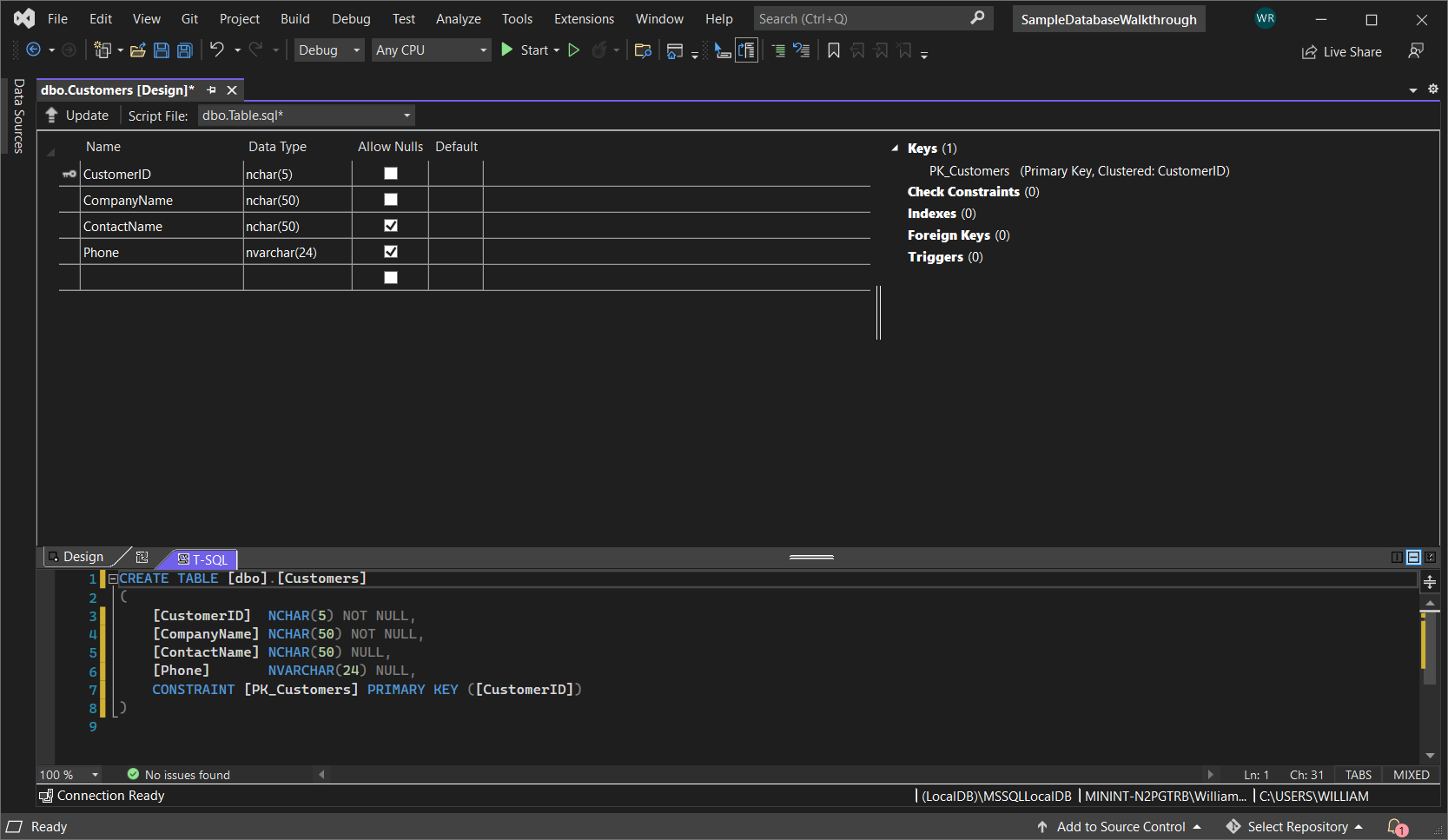
Intellisense при кодирање во Visual Studio

1. **Debugger**- Visual Studio вклучува дебагер кој работи и како дебагер на ниво на извор и како дебагер на ниво на машина. Работи и може да се користи за дебагирање на апликации напишани на кој било јазик поддржан од Visual Studio. Покрај тоа, може да се прикачи и на процесите што се извршуваат, да ги следи и дебагира тие процеси.



3. Debugger на Visual Studio во функција

1. **Data Designer**- алатка за креирање на нови или модификација на веќе постоечки табели. Со неа може да се внесуваат примарни и надворешни клучеви како и граници на табелите.

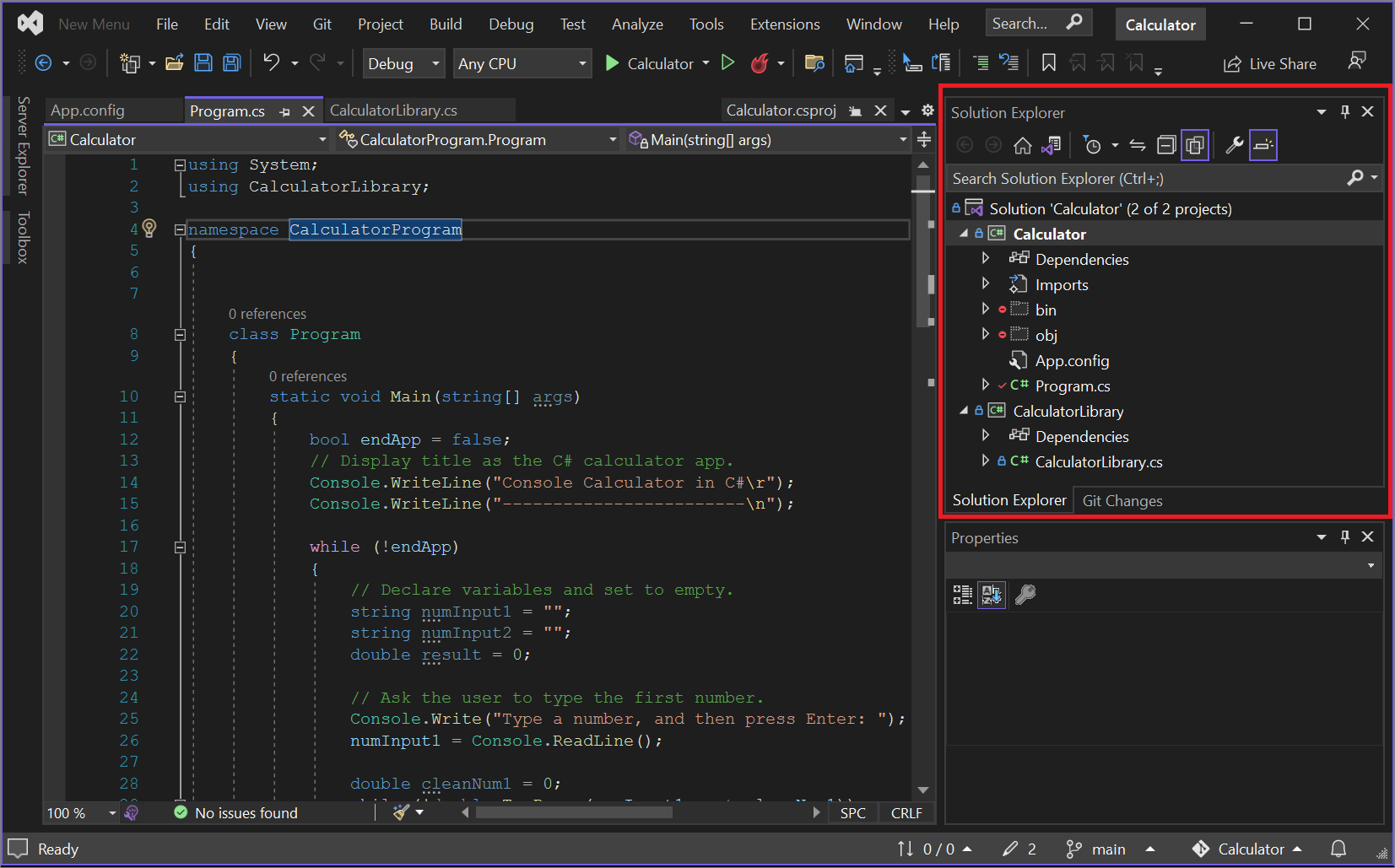
****

4. Интерфејсот на Data Designer

1. **Object Explorer-** обезбедува хиерархиски кориснички интерфејс за прегледување и управување со објектите во секој примерок на SQL Server. Панелот Object Explorer Details претставува табеларен приказ на примероци на објекти и можност за пребарување на одредени објекти. A screenshot of a computer

   Description automatically generated

5. Интерфејс на Object Explorer

1. **Solution Explorer-** Solution е збир на датотеки со код и други ресурси кои се користат за изградба на апликација. Датотеките во решението се распоредени хиерархиски, што може но немора да ја одразува организацијата во датотечниот систем. Solution Explorer се користи за управување и прелистување на датотеките во решението. 6. Интерфејс на Solution Explorer

# Организација и структура на апликацијата

# Најава во апликацијата (Login)

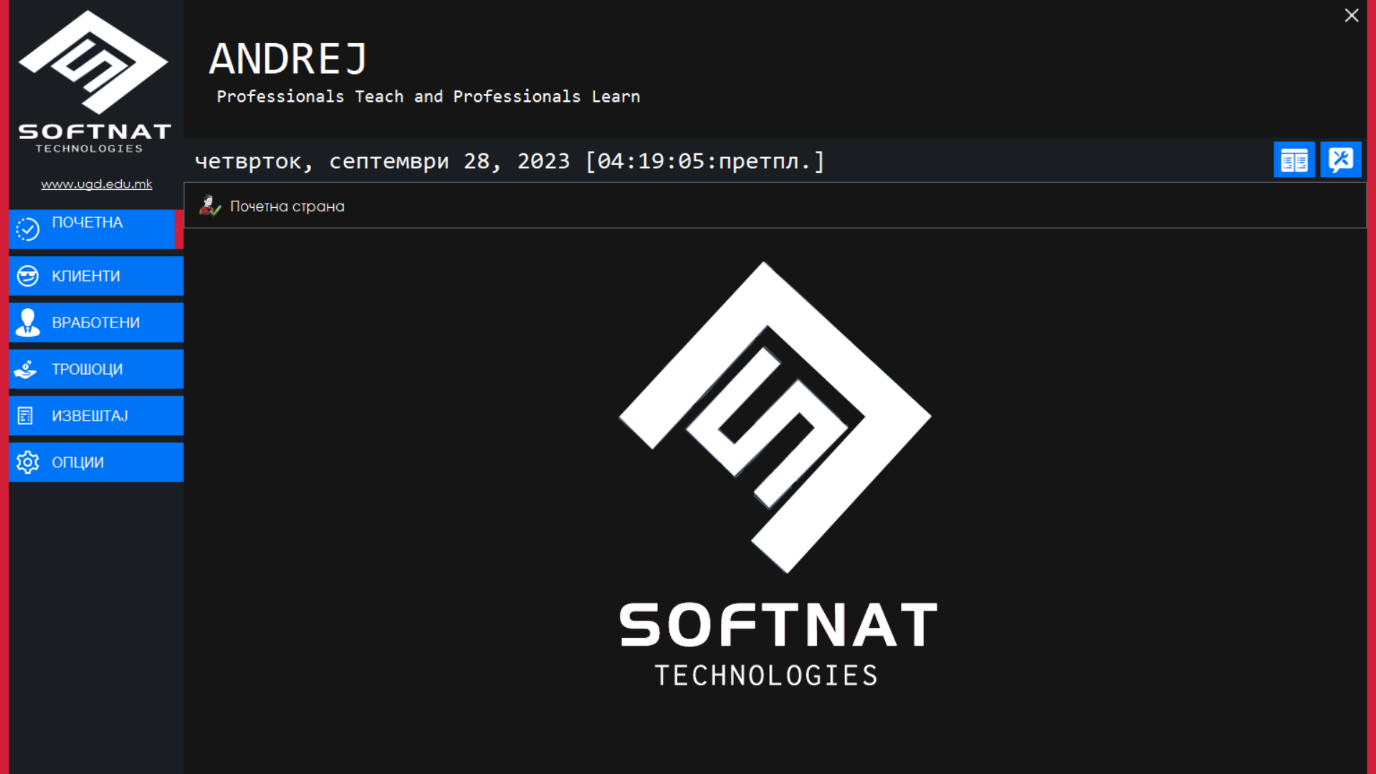
При вклучување на апликацијата прво нешто што ќе се добие е екранот за најавување. Овде се внесуваат корисничко име и шифра на корисникот за да може да се користи апликацијата.



7. Login екран

## Почетна страна (Home Page)

Кога ќе се логираме и влеземе во апликацијата, прво ни се отвара почетната страна каде имаме преглед на сите менија, како и датумот и времето.



1. Почетна страна на апликацијата

## Мени за менаџмент со клиенти

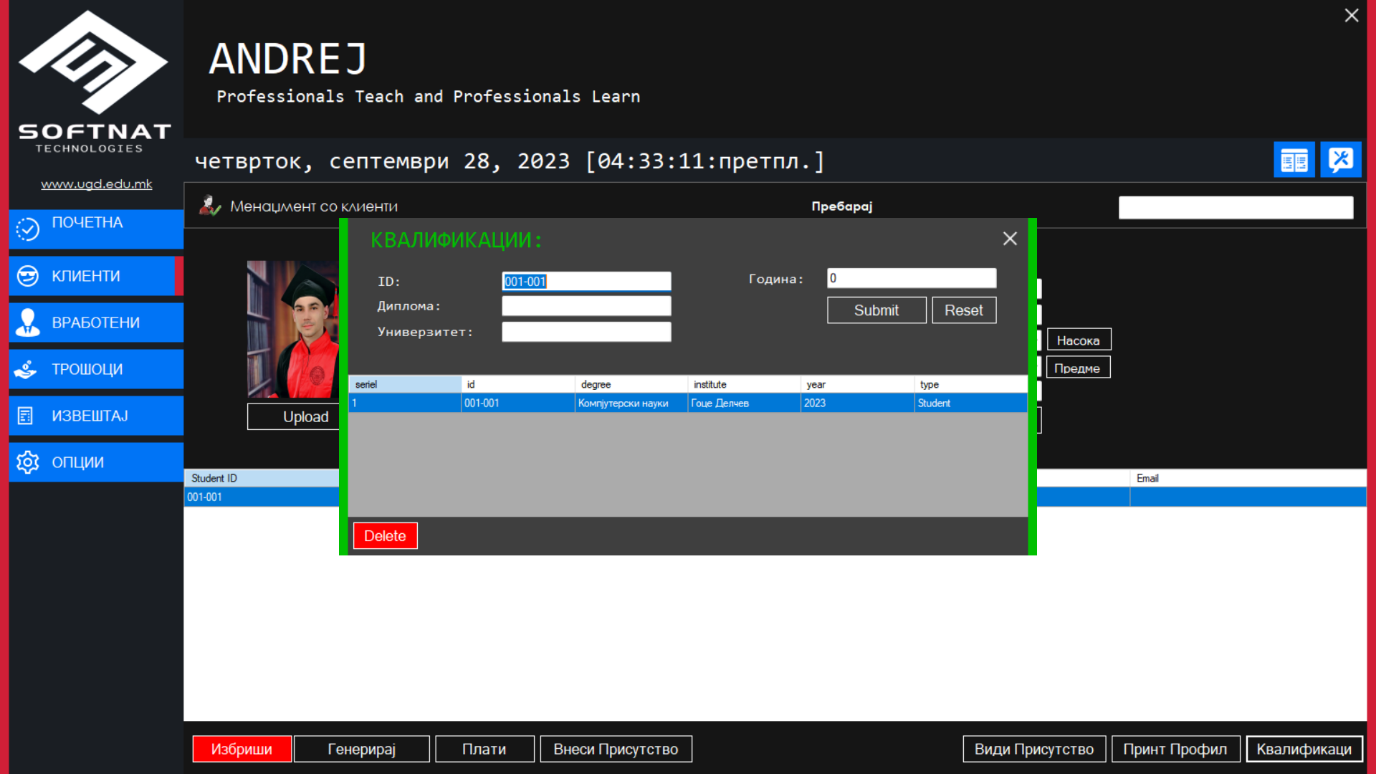
Во менито за менаџмент со клиенти може да се додаваат, бришат како и да се модификуваат постоечки информации за одреден клиент.

A screenshot of a computer

Description automatically generated

1. Додавање на квалификации

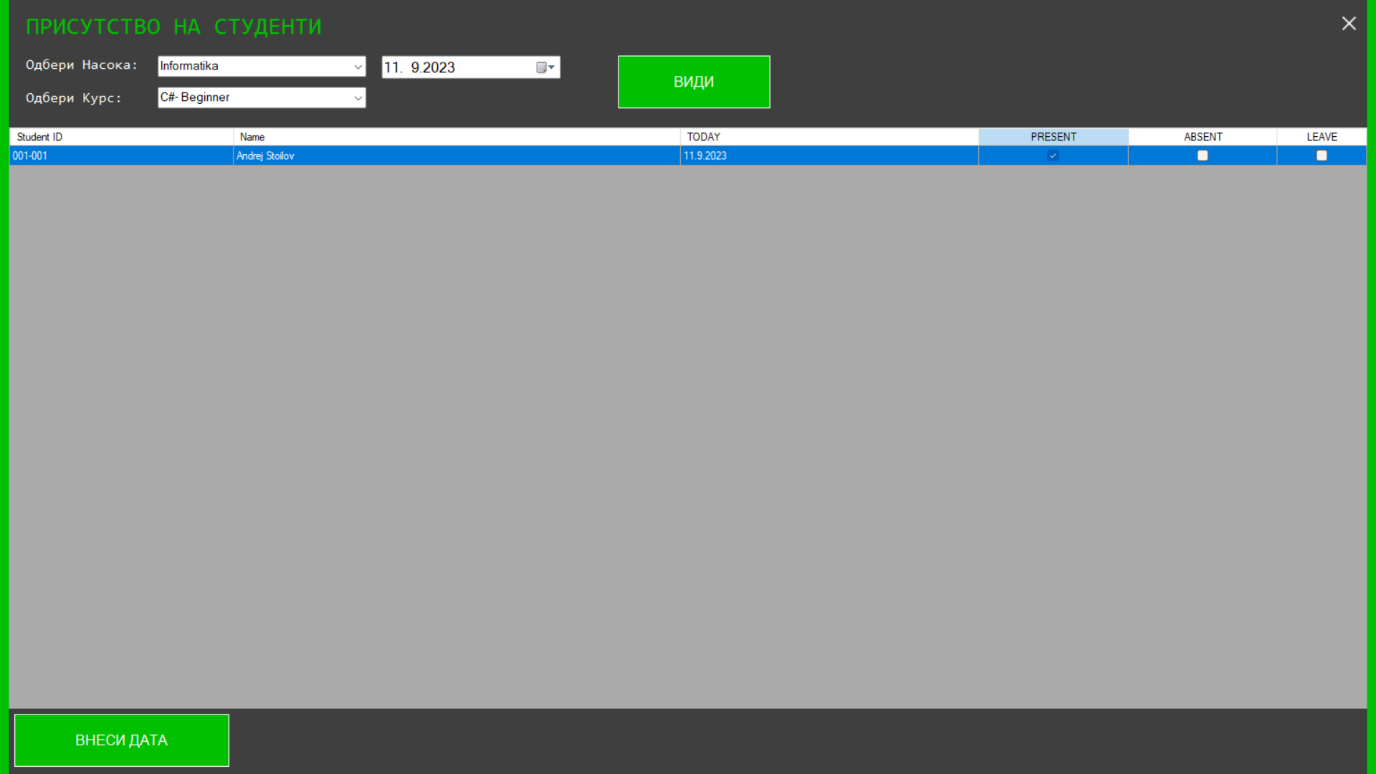
Кога ќе се внесе нов клиент со неговите лични информации, имаме повеќе опции за управување со овој клиент. Можеме да додаваме квалификации доколку клиентот има некакво претходно образование или диплома.



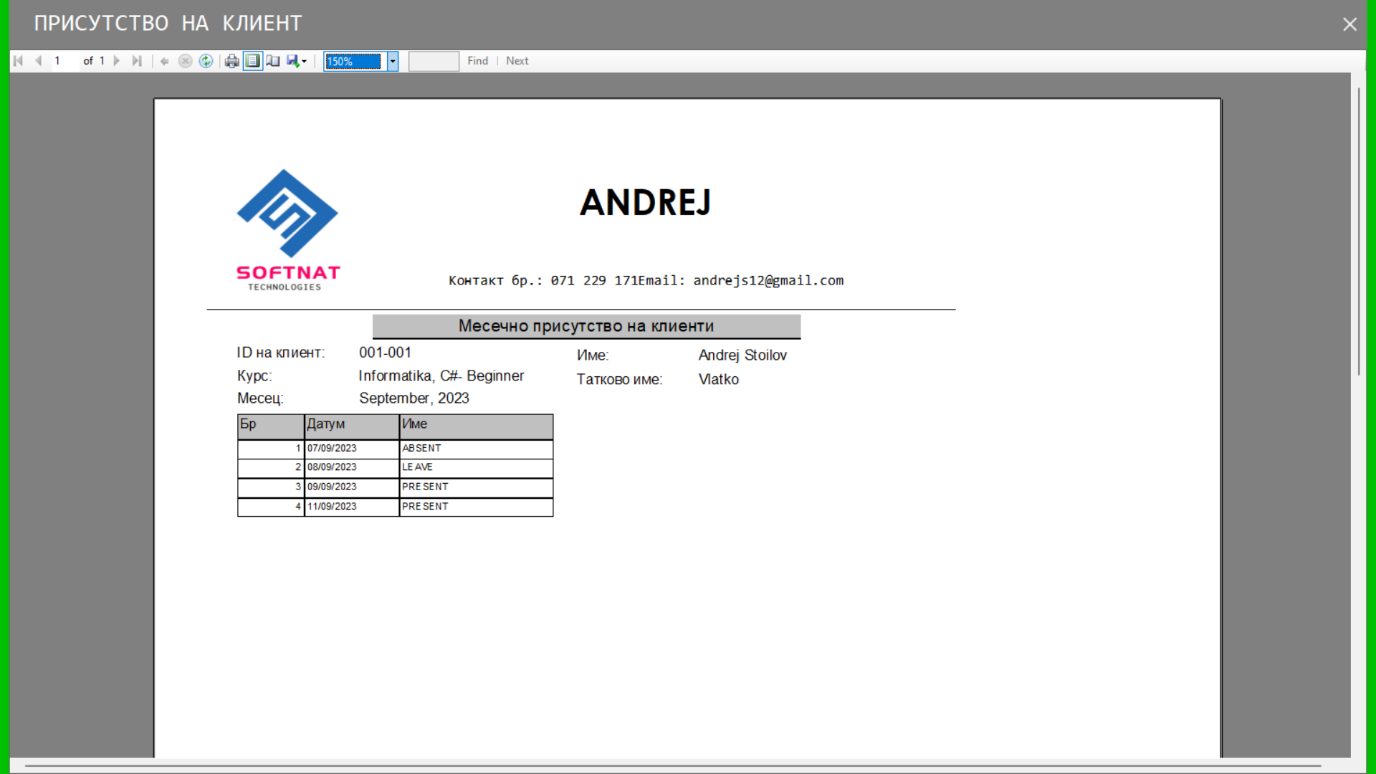
1. Екран за додавање на квалификации

# Преглед и внес на присутство

Доколку имаме клиент кој е запишан на еден или повеќе курсеви, во менито за клиенти имаме опција да водиме евиденција за неговото присутство на тие курсеви и да го прегледаме истото.

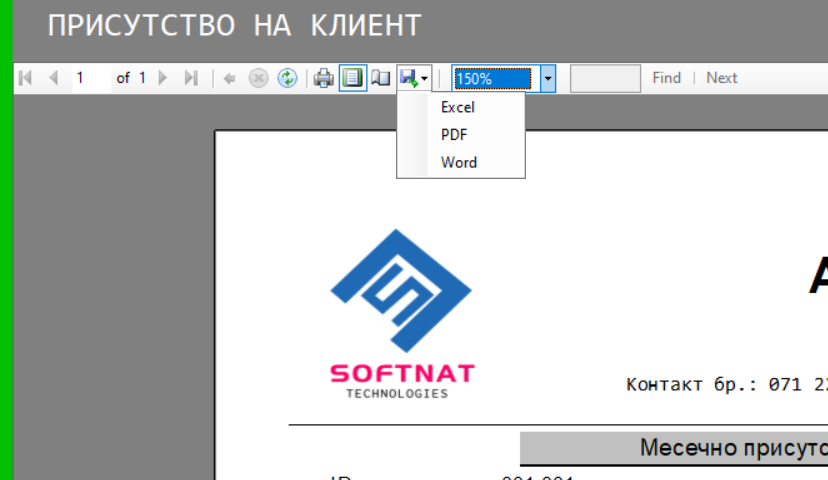


11. Интерфејс за внесување на присутство на клиент



12. Преглед на присутство на клиент за еден месец

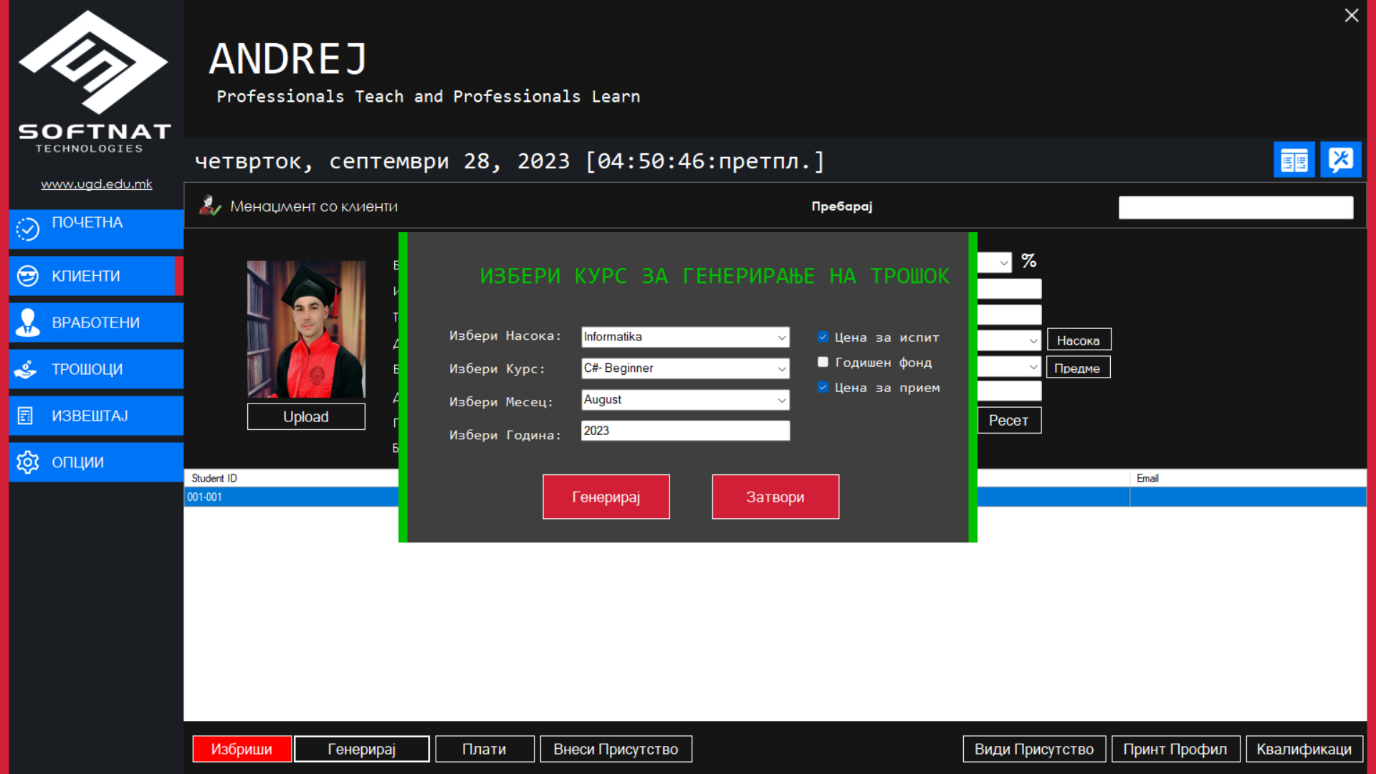
Секој изваден извештај во апликацијата има опција да се извади и зачува како Word, PDF или Excel документ.



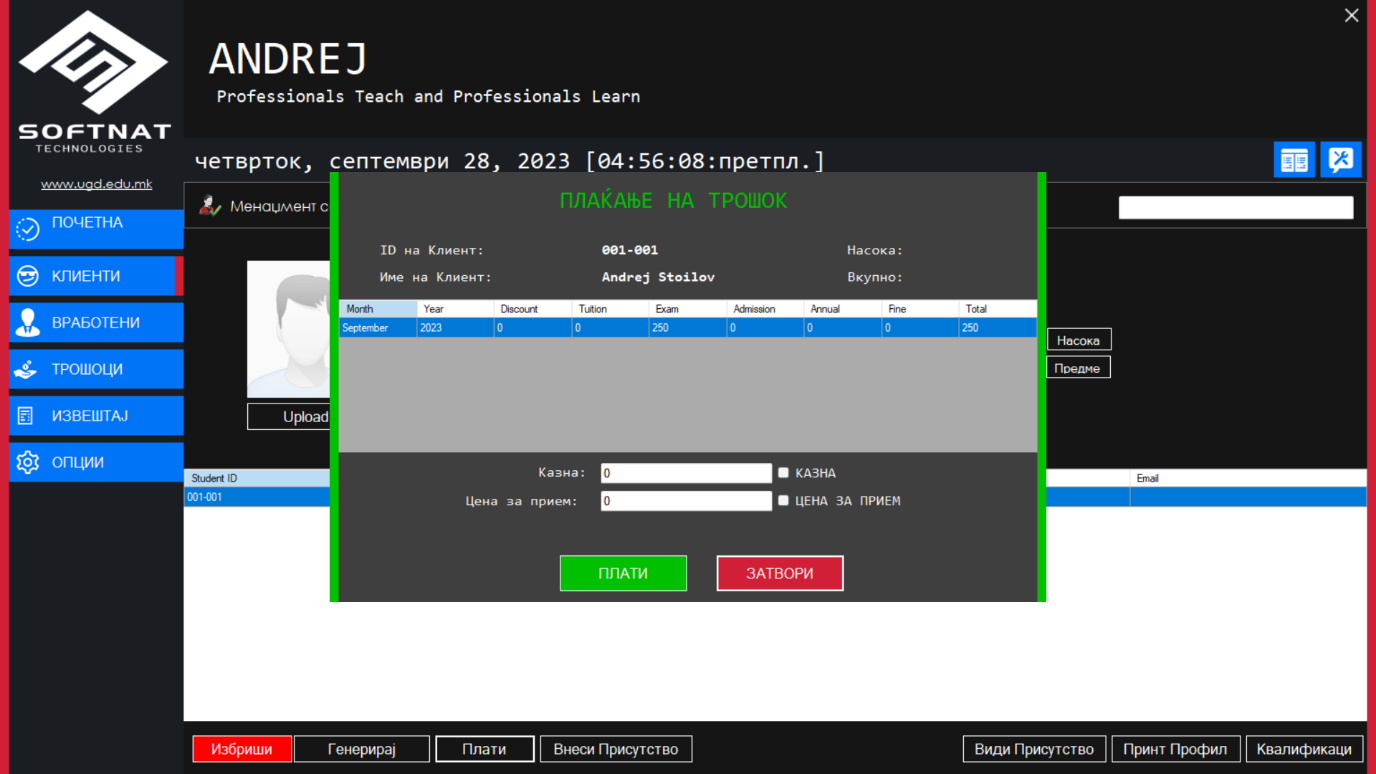
13. Опции за експорт на извештаи

# 4.4 Генерирање и плаќање на трошок на клиент

Со оваа опција можеме да генерираме трошок кој го стекнал еден клиент,тоа може да биде цена за еден испит, цена за прием на курс, и да одбереме дали сакаме наплатеното да излегува од годишниот фонд на клиентот (доколку го има).



14. Генерирање на трошок на клиент за еден месец



15. Плаќање на месечен трошок на клиент

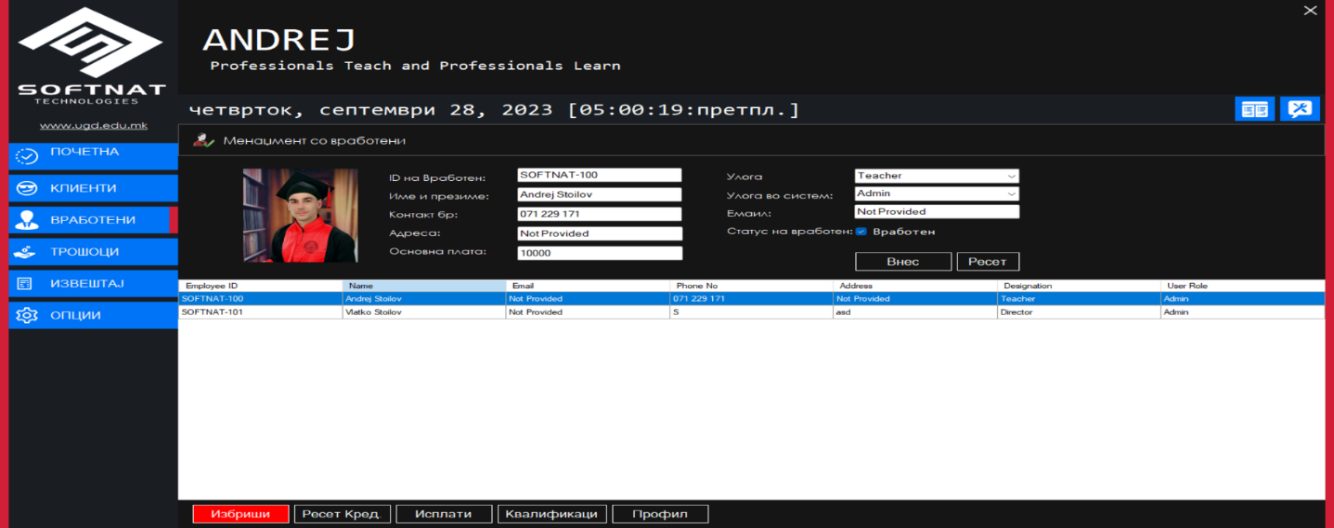
При секоја наплата, штом наплатата ќе се процесира апликацијата генерира 2 потврди од кои една е за клиентот, а другата за центарот за обука.



16. Потврди за наплата на клиент и на центар за обука

## Мени за менаџмент со вработени

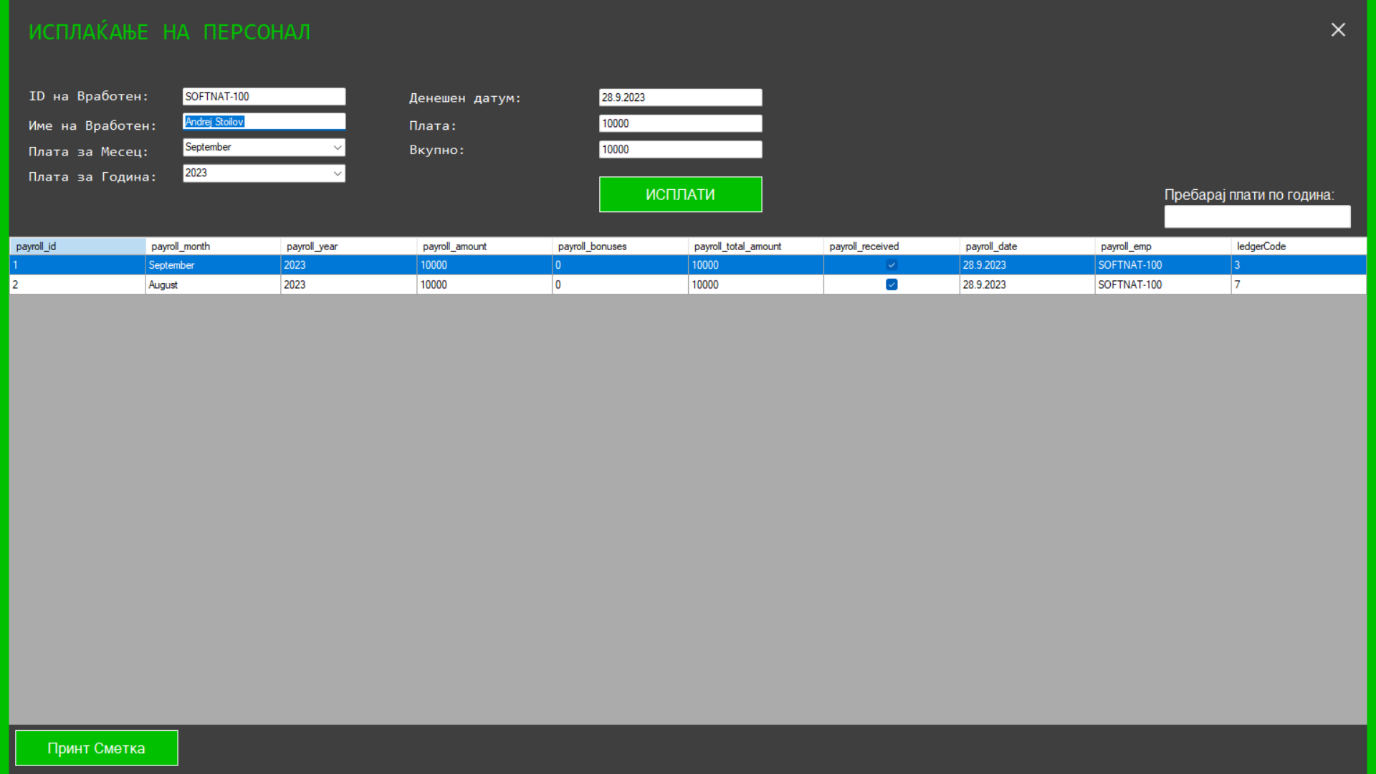
Во менито за менџмент со вработени можеме да внесуваме, бришеме и модифицираме постоечки информации за даден вработен. При внес на вработен, внесуваме ID на вработениот, име и презиме, контакт, адреса, основна плата на вработениот како и улога на вработениот во центарот за обука и улога во самиот систем.



17. Мени за менаџмент со вработени

# 4.6 Исплаќање на вработен

Во менито за вработени постои и опција за исплаќање на вработениот, каде што кога ќе одбереме исплата, треба да се внесе сума на исплата како и за кој месец се исплаќа вработениот.



18. Мени за исплаќање на персонал

При секоја исплата се генерираат сметки за доказ на исплатата кои се достапни за принтање во секое време.

A screenshot of a computer

Description automatically generated

19. Доказ за исплата

## Мени за преглед и внес на трошоци

Во менито за преглед и внес на трошоци се внесуваат дневните или месечните трошоци на центарот за обука. При внесување се внесува износот на трошок, вид на трошок дали е тоа дебит или кредит, причината за трошокот како и краток опис. Овде автоматски излегуваат и платите кои се и платите кои се исплатени на вработените.

A screenshot of a computer

Description automatically generated

20. Дневен внес на трошок

На копчето Види Вкупно, имаме месечен преглед на трошоците за било кој месец во годината.

A screenshot of a computer

Description automatically generated

21. Преглед на месечни трошоци

## Мени за финансиски извештаи

Во менито за финансиски извештаи можеме да добиеме комплетен преглед на финансиите на центарот за обука. Можеме да извадиме дневен, месечен и годишен извештај за било каков трошок или приход на центарот за обука, како и преглед на исплатени плати на вработените.

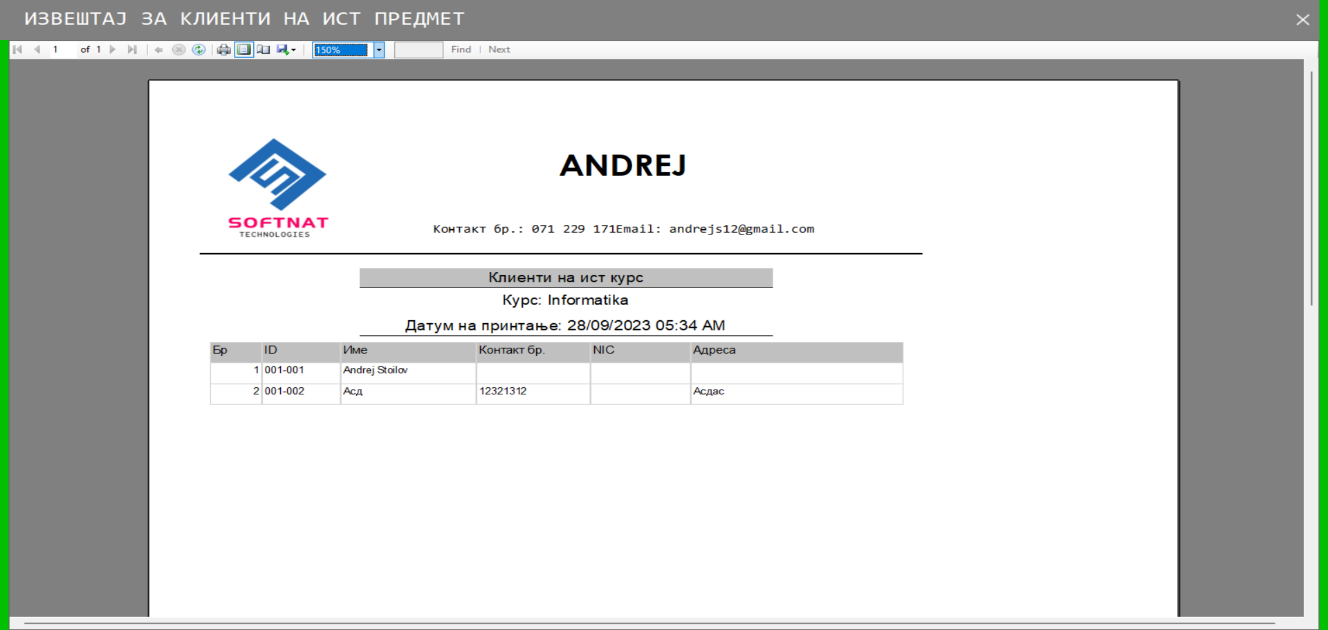
A screenshot of a computer

Description automatically generated

22. Извештај за месечни трошоци

# 4.8.1 Извештаи за клиенти

За клиентите можеме да извадеме моментални листи за клиенти кои се на запишани на ист курс, како и за клиенти кои се на некаква стипендија.



23. Клиенти запишани на ист курс

A screenshot of a computer

Description automatically generated

24. Преглед на стипендирани студенти

# 4.8.2 Извештаи за приход

Во извештаите за приход можеме да го видеме секој внес на средства во центарот за обука од страна на клиентите.

A screenshot of a computer

Description automatically generated

25. Месечен преглед на приход

A screenshot of a computer

Description automatically generated

26. Годишен преглед на приход

# 4.8.3 Извештаи за неплатени трошоци

Доколку сакаме можеме да извадеме месечен извештај за неплатени трошоци на клиентите, а може и да одбереме годишен преглед на неплатените трошоци.

A screenshot of a computer

Description automatically generated

27. Месечен преглед на неплатени трошоци

A screenshot of a computer

Description automatically generated

28. Вкупен годишен извештај за неплатени трошоци

# 4.8.4 Извештаи за месечни плати

Последна опција во менито за извештаи е преглед на исплатени вработени по месец.

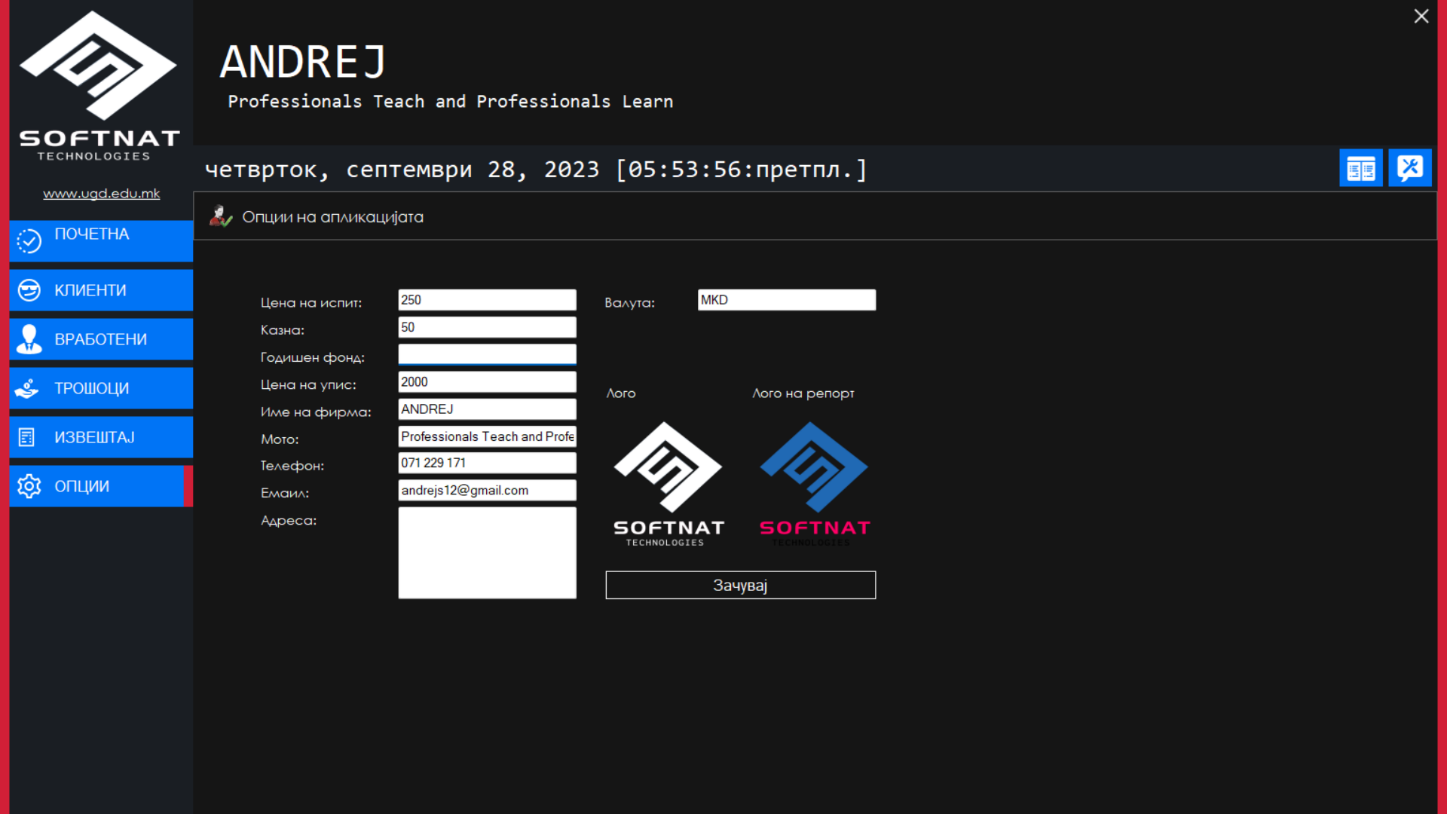
A screenshot of a computer

Description automatically generated

29. Преглед на исплатени плати во Септември

## Опции

Менито на опции е она што ја прави апликацијата лесна за имплементација. Дозволува апликацијата лесно да се адаптира во било која компанија. Можеме лесно да внесеме име и мото на една компанија како и контакт информации. Тука се внесуваат цените на еден испит, цената за упис како и сумата за казна при каснење со уплата, за да не мора при секое плаќање да се внесуваат повторно. Во ова мени исто така можеме лесно да го смениме логото на апликацијата како и логото кое се прикажува при печатење на извештај.



30. Интерфејс на менито Опции

## Мени за промена на информации при најавување

Доколку еден корисник сака да ги промени своите логин информации од било каква причина, само се клика на горното десно плаво копче кое го носи до менито за промена на информации. Тука може да одбере дали сака да промени ID за логин, или сака да ја промени лозинката.

A screenshot of a computer

Description automatically generated

31. Мени за промена на креденцијали

## Мени за Back-Up и Restore на базата на податоци

Најголема грижа во секој систем е безбедноста на информациите кои се наоѓаат во него. Поради ова е направено мени кое дозволува во секој момент брзо и лесно да се креираат Back-Up фајлови од датабазата во моментална состојба, која подоцна може да се врати преку истото мени со кликање на Restore.

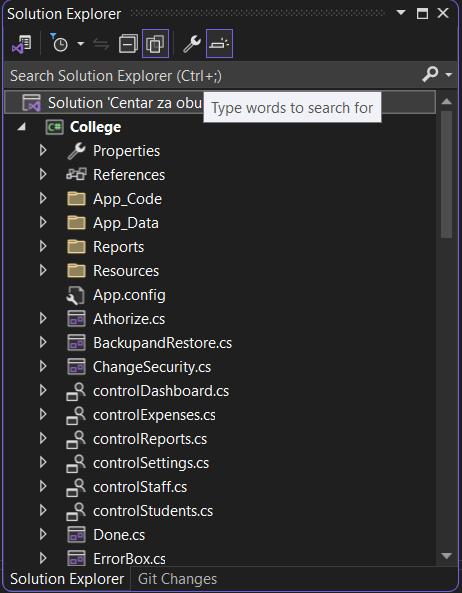
A screenshot of a computer

Description automatically generated

32. Мени за Back-Up и Restore на датабаза

# Структура на датотеки на апликацијата

На сликата е прикажан апликацискиот директориум ,,**Centar za obuka**” отворен во Solution Explorer во **Visual Studio**.



33. Директориум на „Центар за обука“

Тој ги содржи следниве директориуми и датотеки:

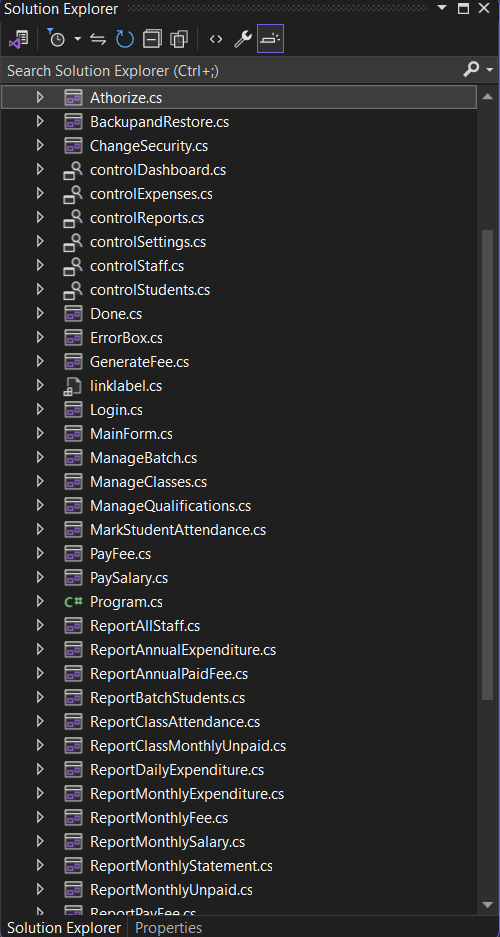
1. **Properties-** опции за вклучување на апликацијата и локација на ресурси и датабаза
2. **References**
3. **App\_Code-**  ги содржи сите back-end .cs датотеки преку кои функционира апликацијата
4. **App\_Data**- ја содржи базата на податоци и сите фајлови поврзани со неа и нејзината функција
5. **Reports**- го содржи дизајнот за сите извештаи кои ги вадиме од апликацијата
6. **Resources**- тука се сите слики искористени во апликацијата
7. **App.config**- конфигурација на .Net Framework и поврзување со датабазата

# Front end

Front end делот се однесува на графичкиот кориснички интерфејст што корисниците го гледаат на веб апликацијата. Овде е користен програмскиот јазик C#.

## C#

C# е современ, универзален програмски јазик развиен од Microsoft. Широко користен за изградба на разновидни апликации, вклучувајќи десктоп апликации, веб апликации и игри. Клучни карактеристики вклучуваат автоматско управување со меморијата и богата стандардна библиотека. C# е познат по својата независност од платформите за работа и е популарен избор за програмери благодарение на неговата универзалност и опширна поддршка за алатки. На сликата подолу се прикажани fron-end датотеките на апликацијата.



34. Front-End датотеки напишани во C#

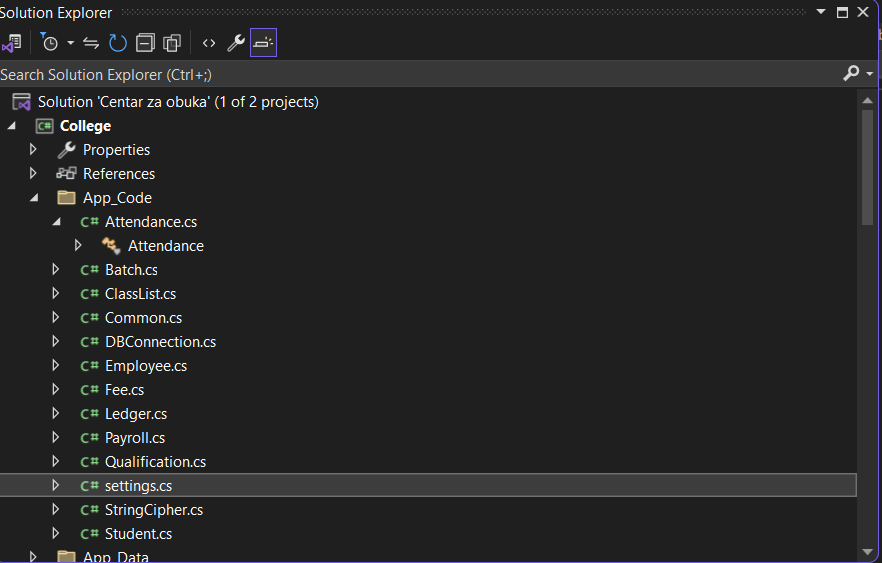
# Back end

Back end делот е изработен во програмскиот јазик **C#** со **.NET** framework.

Најзначајни делови од back end делот се директориумите **App\_Code**, **Properties, Program.cs** и датотеката **App.config**.

## Фолдер App\_Code

Фолдерот App\_Code ги содржи главните фајлови за функционалност на back-end делот на апликацијата.



35. Датотеки во App\_Code

Код од Back-end датотеката Attendance.cs

using System;

using System.Data;

using System.Data.SqlClient;

namespace College.App\_Code

{

class Attendance

{

public int id { get; set; }

public string ids { get; set; }

public DateTime date { get; set; }

public string status { get; set; }

public string type { get; set; }

public string month { get; set; }

public int year { get; set; }

DBConnection strcon = new DBConnection();

//--------------------------------------------------

public bool checkAttendance()

{

SqlConnection connection = strcon.connect();

SqlCommand cmd1 = new SqlCommand();

cmd1.CommandText = "SELECT COUNT(\*) from Attendance where ids='" + this.ids + "' AND date='" + this.date + "'";

cmd1.CommandType = CommandType.Text;

cmd1.Connection = connection;

Int32 count = (Int32)cmd1.ExecuteScalar();

connection.Close();

if (count > 0)

return true;

else

return false;

public string getStatus()

{

SqlConnection connection = strcon.connect();

SqlCommand cmd1 = new SqlCommand();

cmd1.CommandText = "SELECT status from Attendance where ids='" + this.ids + "' AND date='" + this.date + "'";

cmd1.CommandType = CommandType.Text;

cmd1.Connection = connection;

string count = cmd1.ExecuteScalar().ToString();

connection.Close();

return count;

public void update()

{

try

{

SqlConnection connection = strcon.connect();

SqlCommand cmd = connection.CreateCommand();

cmd.CommandText = "update attendance set status=@status where ids='" + this.ids + "' AND date='" + this.date + "'";

cmd.Parameters.AddWithValue("@status", this.status);

cmd.CommandType = CommandType.Text;

cmd.ExecuteNonQuery();

connection.Close();

}

catch (Exception ex)

{

throw new Exception("Error updating teacher attendace");

}

public void insert()

{

try

{

SqlConnection connection = strcon.connect();

string sql = "INSERT INTO Attendance(ids,date,status,type)"

+ "VALUES(@ids,@date,@status,@type)";

SqlCommand cmd = new SqlCommand(sql, connection);

cmd.Parameters.AddWithValue("@ids", this.ids);

cmd.Parameters.AddWithValue("@date", this.date);

cmd.Parameters.AddWithValue("@status", this.status);

cmd.Parameters.AddWithValue("@type", this.type);

cmd.ExecuteNonQuery();

connection.Close();

}

catch (Exception ex)

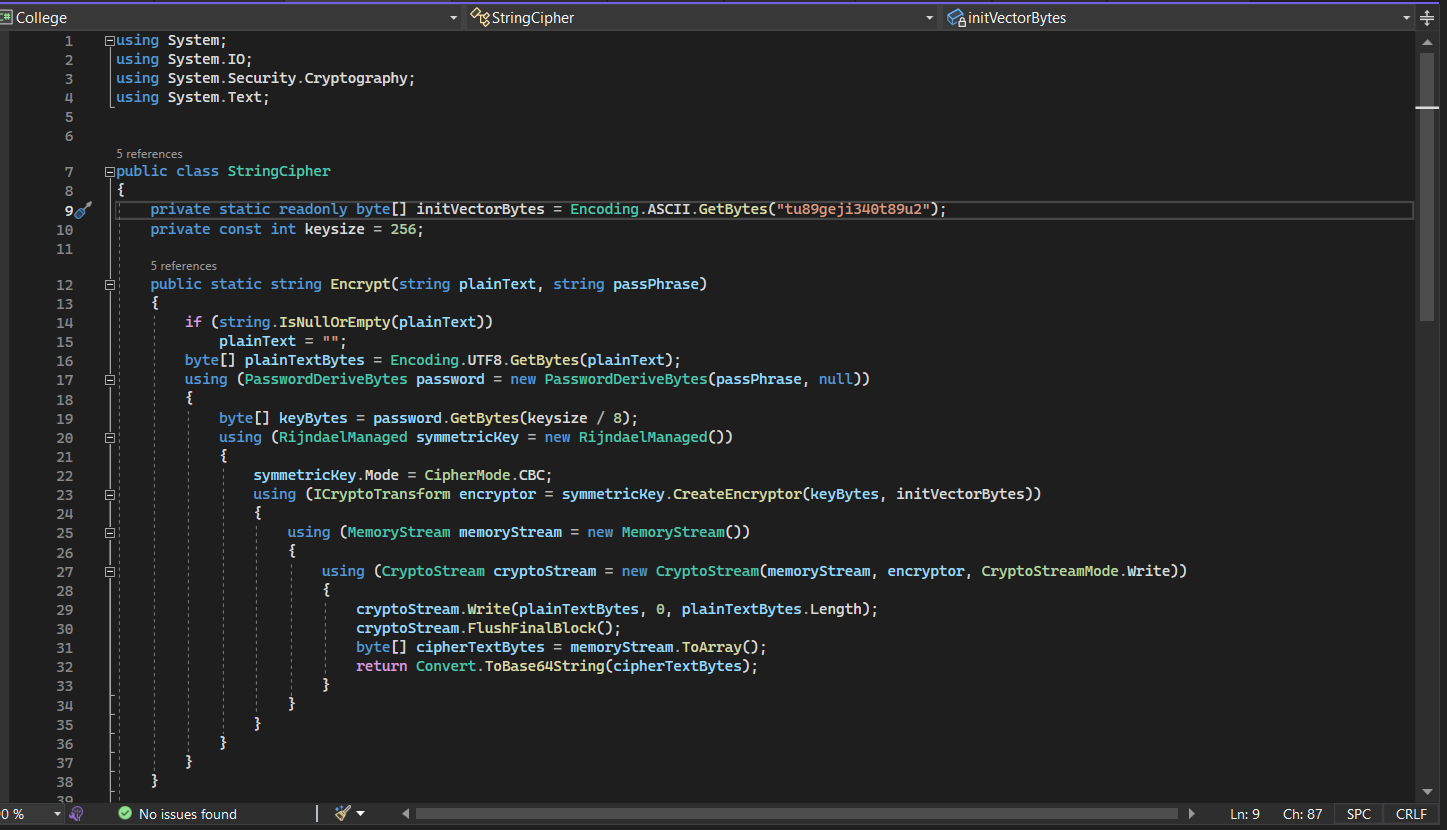
{

throw new Exception("Could not insert attendance");

}

## Систем за автентификација со AES

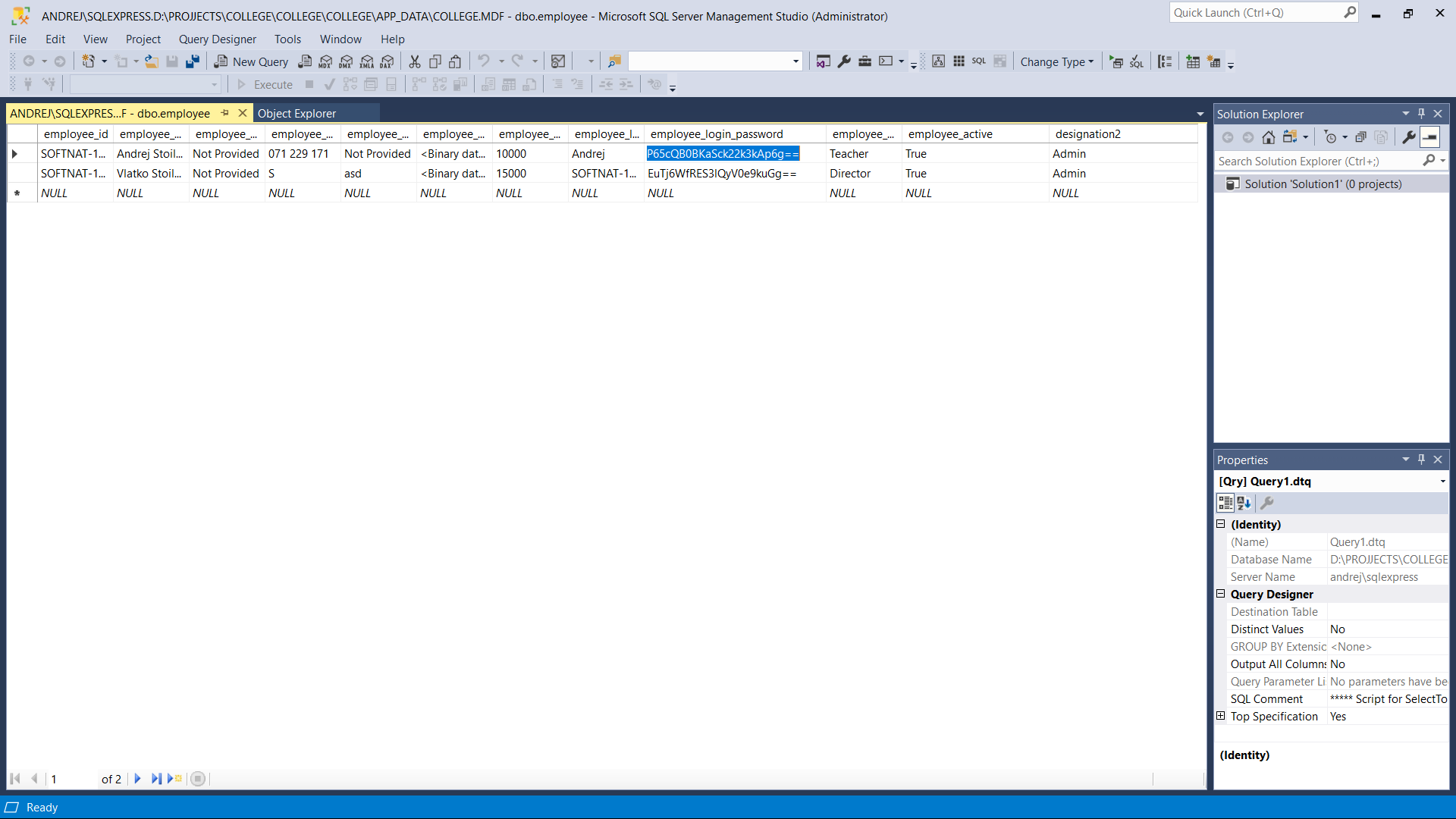
Напредниот стандард за шифрирање (AES) е меѓународно познат алгоритам за симетрично шифрирање познат по својата голема безбедност и ефикасност. Работејќи на блокови на податоци со фиксна големина, AES користи тајни клучеви со должина од 128, 192 или 256 бита и користи софистицирана математичка рамка за извршување на процедури за шифрирање. Се разликува по својата исклучителна отпорност на напади со брутална сила или brute-force attacks. AES е широко распространет и има глобално усвојување во различни апликации, особено во доменот на безбедна комуникација и складирање податоци. Како воспоставен меѓународен стандард за шифрирање, AES нуди различни начини на работа приспособени на специфични случаи на употреба, што го прави незаменлив во доменот на модерната криптографија. AES доаѓа како наследник на Rijndael алгоритамот кој е докажан за стабилен алгоритам.



36. Енкрипција со AES

## Шифрирање на лозинки

Со погорната метода се зема текст како влез и лозинка и враќа кодиран текст како Base64-кодирана низа. Се користи објектот PasswordDeriveBytes за да се изведе клучот од лозинката, потоа се создава RijndaelManaged објект со CipherMode поставено на CBC (Cipher Block Chaining). Се кодира текстот користејќи го изведениот клуч и иницијализациониот вектор, и на крај се враќа кодираниот текст како Base64-кодирана низа. Овој метод обезбедува едноставен начин за кодирање и декодирање на текстови користејќи Rijndael кодирање, кој е безбеден симетричен алгоритам за кодирање. Важно е да се забележи дека безбедноста на кодирањето зависи од јачината на лозинката и внимателното управување со иницијализациониот вектор.



На сликата горе е прикажана табелата Employee од базата на податоци во Microsoft SQL Server Management Studio . Лозинките на сите корисници се исти, но нивните шифрирани верзии се различни.

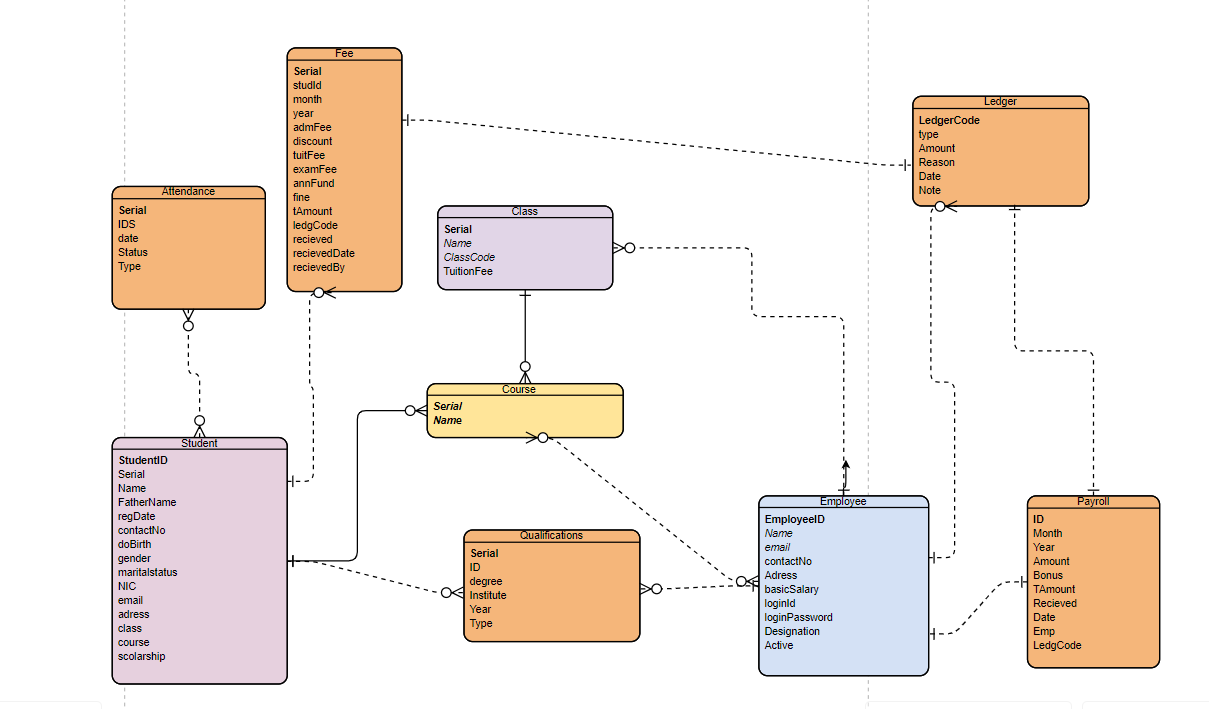
# Креирање на базата на податоци

Во апликацијата се одлучив за MySQL, бидејќи е релациона база на податоци, записите имаат иста структура и полесно се работи со сложени прашалници. SQL базите на податоци се лесно скалабилни вертикално, што значи дека лесно се додаваат нови записи, но тешко се додаваат или уништуваат атрибути, бидејќи тоа ќе треба да се направи за секој запис во табелата. Компатибилен е со сите платформи како Windows, MacOS, Linux, има голема онлајн поддршка за било какви проблеми и одлична безбедност.

## ERD(Entity relationship diagram) дијаграм

На сликата долу е прикажан **ERD(Entity relationship diagram)** дијаграм, каде плавите правоаголници се ентитетите, заедно со нивните атрибути.

Дијаграмот е изработен во **Visual Paradigm Online**.



37.Базата на податоци е нормализирана до 4-та нормална форма.

Во продолжение се напишани табелите(ентитетите) и нивните атрибути, каде што атрибутите означени со **#** пред нив, се **примарни клучеви**, а оние означени со **\*** се **надворешни клучеви**.

**Attendance** (#id, ids, date, status, type, \*PK\_Attendance)

**Course** (#serial, batchName, \*PK\_Course)

**Class** (#serial, className, classCode, tuitionFee, \*PK\_Class)

**Employee** (#employee\_id, employee\_name, email, employee\_phone\_number, employee\_adress, employee\_picture, employee\_basic\_salary, employee\_login\_id, employee\_login\_password, employee\_designation, employee\_active, \*PK\_Employee)

**Fee** (#serial, studentID, month, year, admissionFee, discount, tuitionFee, examFee, annualFund, fine, totalAmount, received, recievedDate, recievedBy, ledgerCode, \*PK\_Fee)

**Ledger** (#ledgerCode, type, amount, reason, date, note, \*PK\_Ledger)

**Payroll** (#payroll\_ID, payroll\_month, payroll\_year, payroll\_amount, payroll\_bonuses, payroll\_total\_amount, payroll\_recieved, payroll\_date, payroll\_emp, ledgerCode, \*PK\_payroll )

**Qualification** (#serial, id, degree, institute, year, type, \*PK\_Qualificatons)

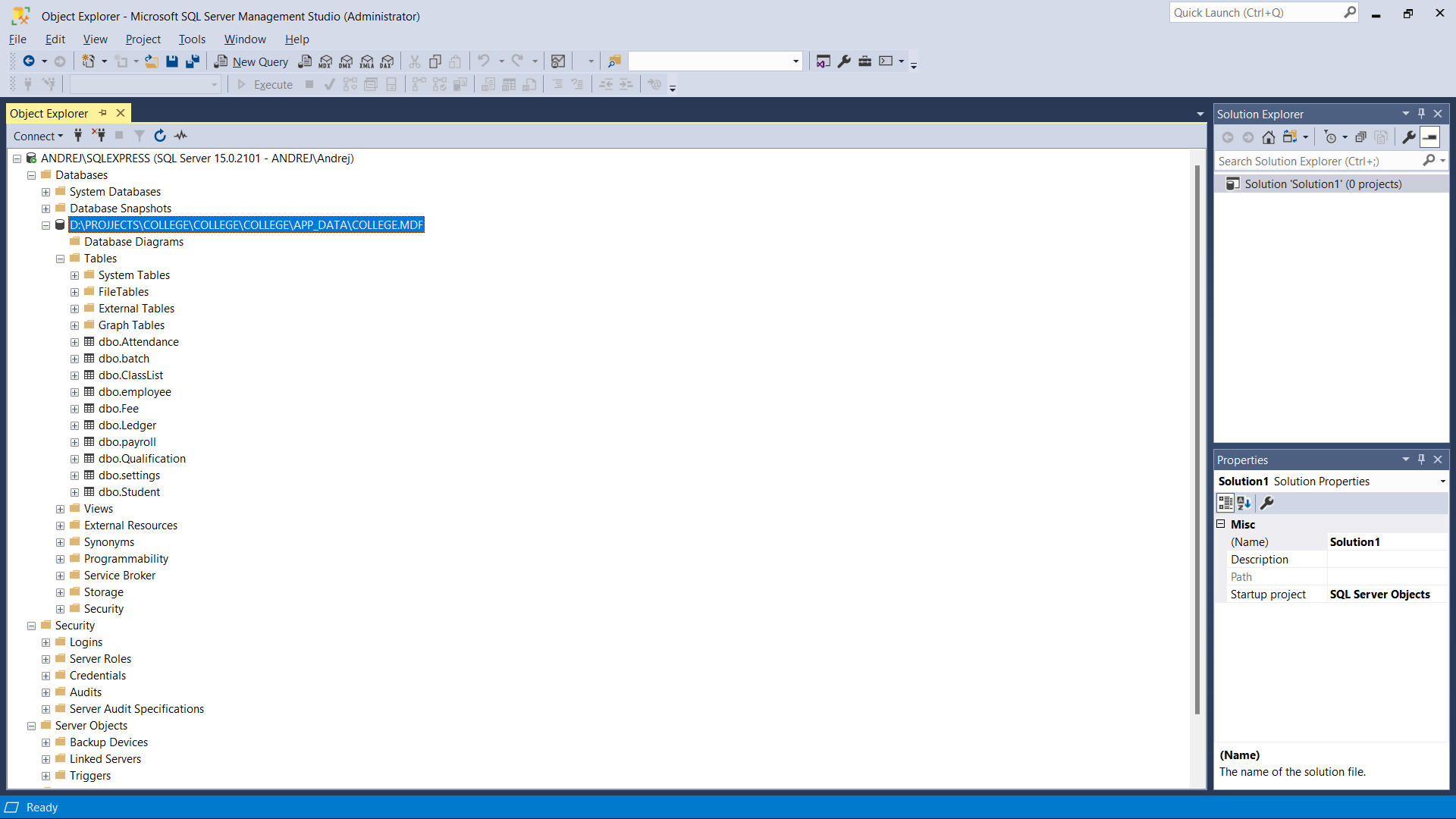
**Settings** (#id, examFee, fine, annualFund, admissionFee, address, phone, email, institute, moto, image, imageReport, currency, \*PK\_Settings)

**Student** (#studentID, serial, name, fatherName, registrationDate, contact, dob, gender, maritalStatus, NIC, email, address, image, batchCode, classCode, scholarship, \*PK\_Student)

**Конфигурирање на базата на податоци:**

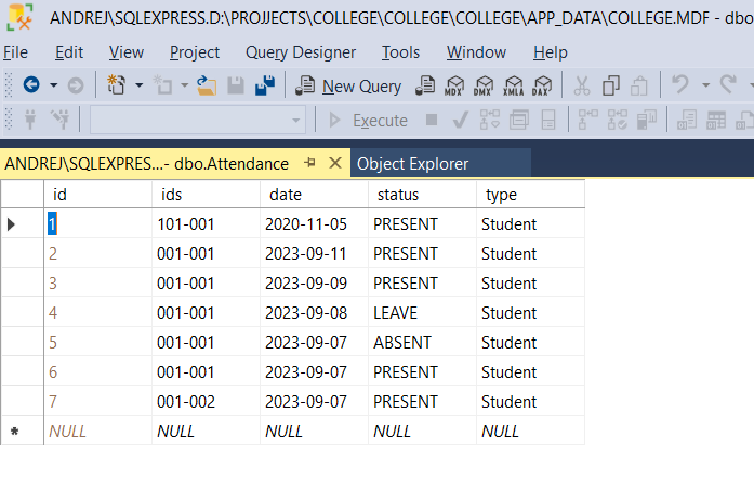
|  |
| --- |
| using System.Data.SqlClient;  namespace College.App\_Code  {  class DBConnection  {  SqlConnection connection;  public DBConnection()  {  connection = new SqlConnection(@"data source=(local)\SQLEXPRESS;database=D:\PROJJECTS\COLLEGE\COLLEGE\COLLEGE\APP\_DATA\COLLEGE.MDF; integrated security = SSPI");  }  public SqlConnection connect()  {  connection.Open();  return connection;  }  public void disconnect()  {  connection.Close();  }  }  } |

## Структура на табелите во Microsoft SQL Server Management Studio



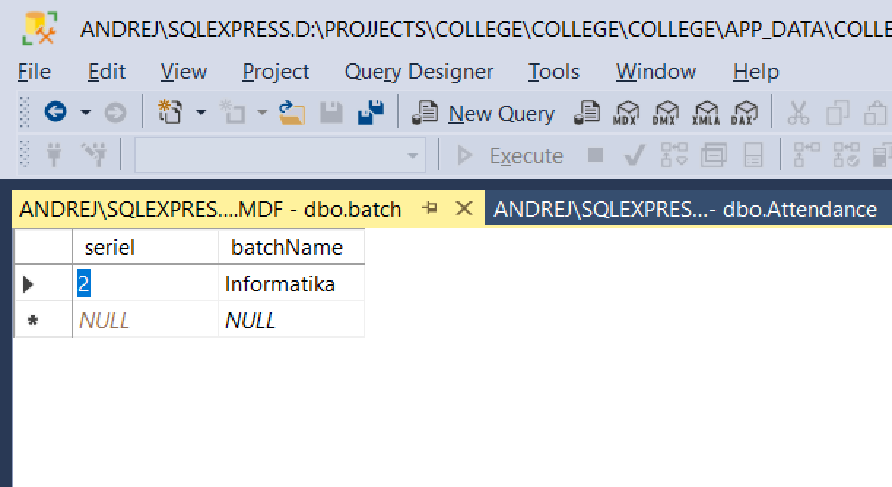
38. Датабазата прикажана во Microsoft SQL Server Management Studio

Структура на табелата **Attendance**:



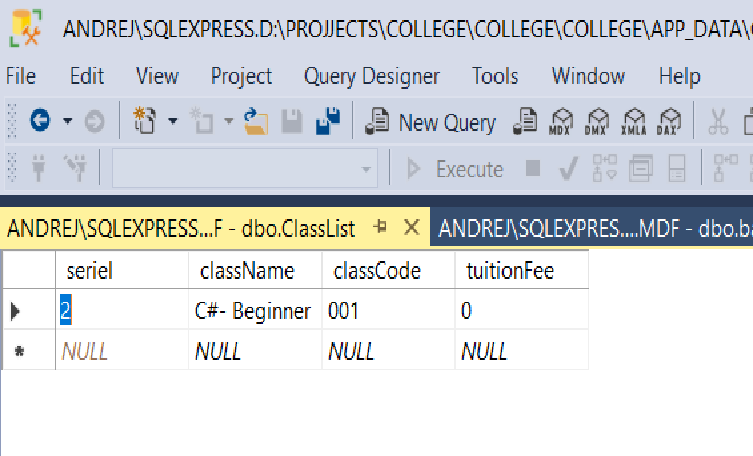
39

Структура на табелата **Batch(Course)**:



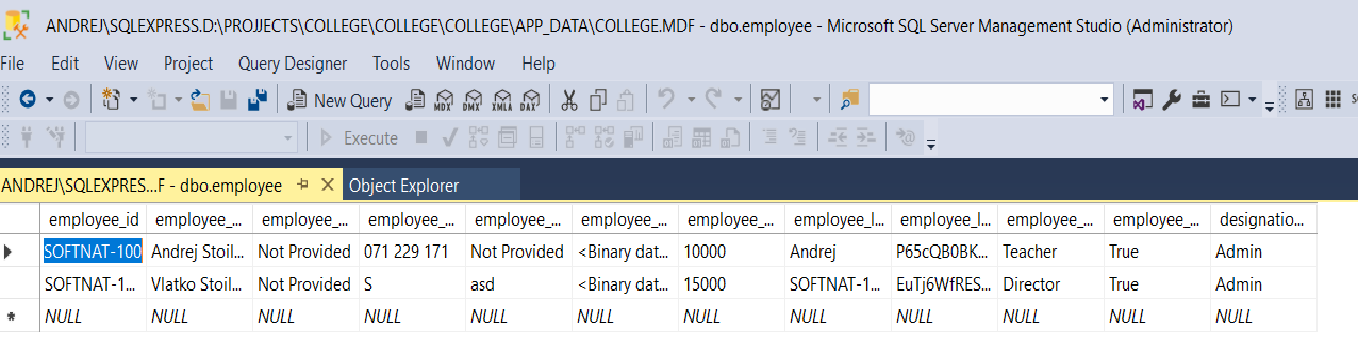
40.

Структура на табелата **Class**:



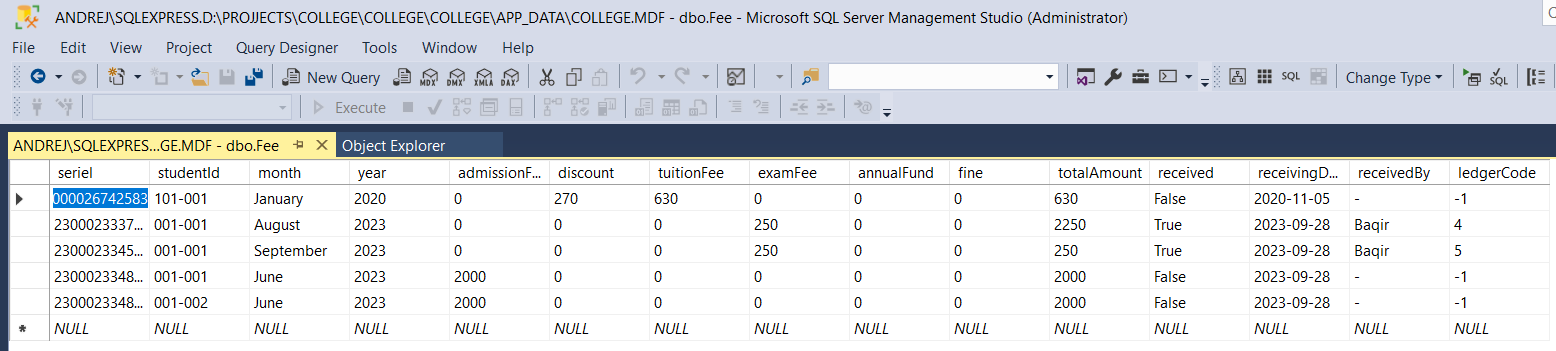
41.

Структура на табелата **Employee**:



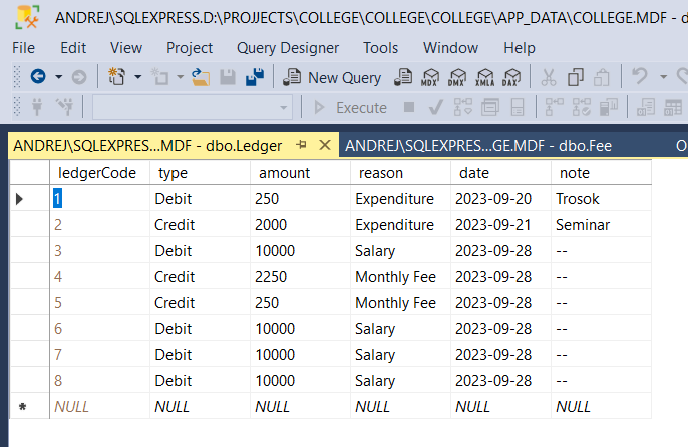
42.

Структура на табелата **Fee**:



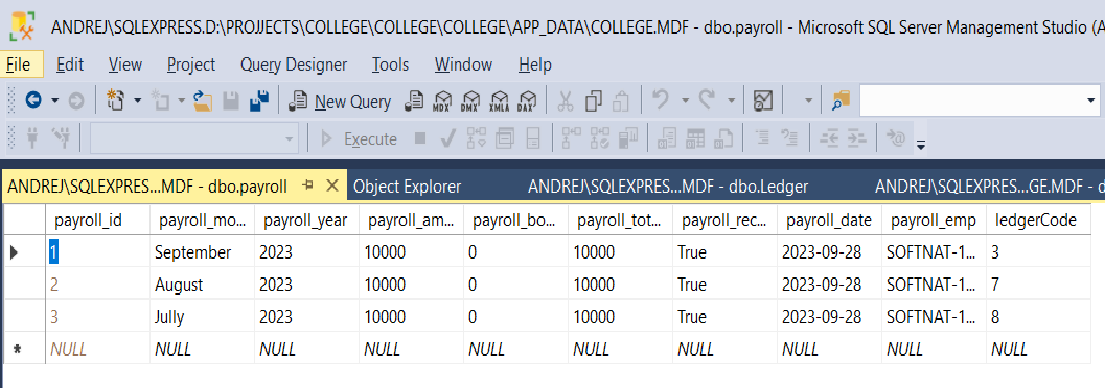
43.

Структура на табелата **Ledger**:



44.

Структура на табелата **Payroll**:



45.

# Заклучок

Овој C# проект за управување со институции посветени на обуки за поттикнување на поединечниот напредок претставува сеопфатно решение. Преку оваа апликација, се стремиме да го поедноставиме управувањето со процесите кои се одвиваат во институциите посветени на испорака на висококвалитетно образование. Нашиот систем нуди ефикасен пристап за управување со курсеви, упис на клиенти, координација на интституцијата, финансиски надзор и извештај. Овој C# проект за управување со институции за професионално усовршување претставува сеопфатен и лесен систем за управување со сите аспекти од работењето на институцијата. Системот е скалабилен, адаптирачки и лесен за употреба. Истиот е тестиран, потврден и подготвен е за користење во производствени средини. Успехот на нашиот проект лежи во неговиот капацитет да ги трансформира институциите во центри на извонредност, поттикнувајќи култура на постојано унапредување. Со обезбедување на приспособливост се грижи за институциите од различни размери, зајакнувајќи ги нивните способности да поттикнат професионален раст. Заклучно, нашиот C# проект за управување со институциите во областа на професионалното усовршување означува витален чекор кон модернизирање и оптимизирање на процесите што го поттикнуваат образованието и професионалниот развој. Уверени сме дека тоа ќе им овозможи на институциите да напредуваат во дигиталната ера, на крајот придонесувајќи за успехот на безброј поединци кои сакаат да ги унапредат своите вештини и експертиза.

# Користена литература

**Front end:**

1. https://www.youtube.com/@IAmTimCorey

**Back end:**

1. https://www.youtube.com/@IAmTimCorey
2. https://www.youtube.com/@freecodecamp
3. https://www.youtube.com/@programmingwithmosh

**Дополнителни ресурси:**

1. <https://www.geeksforgeeks.org/>
2. <https://www.w3schools.com/>
3. <https://stackoverflow.com/>
4. <https://github.com/>
5. <https://learn.microsoft.com/en-us/>
6. <https://moodle2.ugd.edu.mk/>
7. <https://e-lib.ugd.edu.mk/naslovna.php>
8. <https://online.visual-paradigm.com/>